Primjer modeliranja podataka poslovnog procesa skladišta

Gržinić, Nikola

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:617895

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Image not found or type unknown

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

zir.nsk.hr

Image not found or type unknown

Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
«Dr. Mijo Mirković»

Nikola Gržinić

PRIMJER MODELIRANJA PODATAKA POSLOVNOG PROCESA SKLADIŠTA

Završni rad

Pula, 2016.

Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
«Dr. Mijo Mirković»

Nikola Gržinić

PRIMJER MODELIRANJA PODATAKA POSLOVNOG PROCESA SKLADIŠTA

Završni rad

JMBAG: 0303036930 , redoviti student

Studijski smjer: Poslovna informatika

Predmet: Baze podataka

Mentor: prof.dr.sc. Vanja Bevanda

Pula, veljača 2016.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za prvostupnika _____ovime izjavljujem da je ovaj Završni rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student:

_____

U Puli, _____, _____ godine

IZJAVA

o korištenju autorskog djela

Ja, _____ dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom _____

_____

koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama. Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis:

_____

SADRŽAJ:

UVOD

1. BAZA PODATAKA I MODELIRANJE PODATAKA 1
1.1. Koraci i poslovi pri oblikovanju baze podataka 2
2. KONCEPTUALNO MODELIRANJE PODATAKA 4
2.1. Osnovni konstrukti modela entiteti-veze 4
2.1.1. Entiteti 4
2.1.2 Atributi 5
2.1.3. Veze 5
2.2. Napredni konstrukti modela entiteti-veze 11
2.2.1. Generalizacija 11
2.2.2. Agregacija 11
2.3. Izrada modela entiteti-veze 12
3. LOGIČKO MODELIRANJE PODATAKA 13
3.1. Relacijski model podataka 13
3.1.1. Struktura relacijskog modela podataka 15
3.1.2. Ograničenja 19
3.1.3. Operacije 20
3.1.4. Pretvorba konceptualnog modela (entiteti-veze) u logički (relacijski) model 26
3.1.5. Normalizacija 29
3.1.6. SQL 30
4. PRIMJER MODELIRANJA PODATAKA POSLOVNOG PROCESA 32
4.1. Konceptualni model (entiteti-veze) skladišta 32
4.2. Relacijski model skladišta 34
4.3. Rezultati upita 35
5. ZAKLJUČAK 37
6.LITERATURA 38
7. POPIS SLIKA 39

UVOD

Informacije su jedan od najvažnijih resursa u poslovnom svijetu. Jednako je važan i informacijski sustav koji prikuplja, obrađuje i distribuira podatke, kojemu je ujedno i cilj osigurati sve potrebne informacije za obavljanje poslova i pravovremeno donošenje odluka u organizacijama. Može se logički zaključiti od koje su važnosti baze podataka. Baze podataka omogućuju lakše razvrstavanje, te bolju dostupnost podacima, stoga se može reći da je baza podataka zbirka zapisa pohranjenih na računalu. Podaci su dostupni svim korisnicima i aplikacijskim programima, samim time omogućuju brže, bolje i efikasnije poslovanje. Kada se govori o bazama podataka prva pomisao je na relacijske baze podataka upravo zbog svoje zastupljenosti. U relacijskom modelu podaci i veze među podacima prikazani su u tablicama koje se sastoje od redaka i stupaca. Ipak prije same izrade relacijskog modela potrebno je izraditi konceptualni model u kojem se prikupljaju informacijski zahtjevi od strane korisnika. Nakon prikupljenih podataka izrađuje se model entiteti-veze gdje se utvrđuju entiteti, atributi koji opisuju entitete, te veze između entiteta. Prilikom prikupljanja podataka potrebno je imati na umu koja će biti uloga tih podataka u budućoj bazi, te da isti moraju biti povezani i međusobno se nadopunjavati.

Završni rad sastoji se od četiri poglavlja koji obuhvaćaju sve bitne informacije i primjere glede modeliranja podataka poslovnog procesa za skladište. U prvom poglavlju spominje se općenito o bazama podataka kako bi se shvatila bitna materija što su one uopće. Nakon toga objašnjeno je modeliranje podataka, te podjela modela podataka. Drugo poglavlje kreće od konceptualnog modeliranja podataka gdje su objašnjeni osnovni konstrukti modela entiteti-veze: entiteti, atributi i veze, te napredni konstrukti modela entiteti-veze: generalizacija i agregacija. Zatim je opisana izrada modela entiteti-veze. Treće poglavlje govori o logičkom modeliranju podataka. Detaljno je opisan relacijski model podataka, koji se dijeli na tri dijela: struktura u koju spadaju domena, relacije, atributi i ključevi (primarni i strani ključ), ograničenja: entitetni integritet i referencijalni integritet, operacije: unija, presjek, razlika, Kartezijev produkt, selekcija, projekcija, spajanje i dijeljenje. Opisana je pretvorba modela entiteti-veze u relacijski model, normalizacija u kojoj se spominju tri najkorištenije normalne forme, a one su: prva, druga i treća normalna forma, te na kraju SQL jezik koji je u današnje vrijeme jedan od najraširenijih upitnih jezika za rad s relacijskom bazom podataka. Četvrto poglavlje je ujedno i ključno poglavlje gdje je prikazan primjer modeliranja podataka poslovnog procesa skladišta koje je ujedno i tema ovog završnog rada.

1. BAZA PODATAKA I MODELIRANJE PODATAKA

U najgrubljim crtama iskazano, baza podataka predstavlja skup datoteka, organiziranih tako da budu ujednačene (unificirane) s obzirom na strukturu, te povezanih tako da uključuju minimalnu redundanciju podataka i omogućuju korisnicima pristup podacima uz minimalna ograničenja. Dakle, koncept baze podataka evoluira iz koncepta datoteka podataka, odnosno predstavlja njegovu nadogradnju. Ujednačenost (unificiranost) strukture baze podataka ostvaruje se primjenom usklađenih (kompatibilnih) modela podataka na temelju kojih se strukturiraju datoteke što čini bazu podataka.[1]

Baza podataka omogućava lakše razvrstavanje, te bolju dostupnost podacima, stoga se može reći da je baza podataka zbirka zapisa pohranjenih na računalu. Podaci su dostupni svim korisnicima i aplikacijskim programima. Upisivanje, brisanje i čitanje podataka obavlja se preko posebnog softvera tzv. sustava za upravljanje bazom podataka (DBMS-a).

Sustav za upravljanje bazom podataka je poslužitelj (server) baze podataka. Oblikuje fizički prikaz baze u skladu s traženom logičkom strukturom. Također, on obavlja u ime klijenta sve operacije s podacima. U stanju je podržati razne baze, od kojih svaka može imati svoju logičku strukturu, no u skladu s istim modelom. Brine se za sigurnost podataka, te automatizira administrativne poslove s bazom.[2]

Model podataka je simbolički prikaz vremensko – logičkih i hijerarhijskih odnosa među događajima u nekome sustavu, iskazanih podacima kao njihovim manifestacijama. Modeliranje podataka je postupak izrade stanovitog modela podataka i obavlja se tijekom procesa razvitka informacijskog sustava.[3]

Ovisno o potrebama što se javljaju u pojedinim fazama razvitka informacijskog sustava, razlikuju se tri vrste modela podataka:[4]

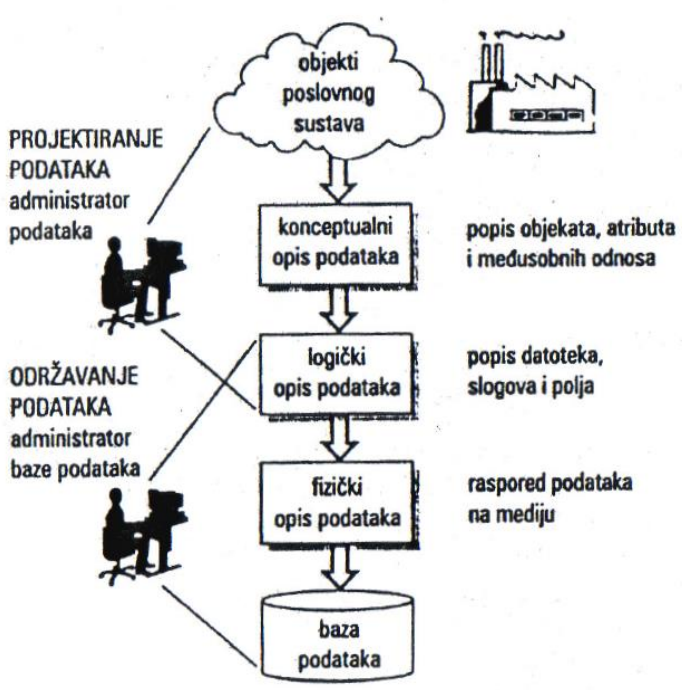
1. Konceptualni model podataka – odražavaju način na koji korisnici budućeg informacijskog sustava shvaćaju događaje u sustav, pa onda i podatke koji proizlaze iz tih događaja.

[1] Panian Željko, Poslovna informatika za ekonomiste, Potecon, Zagreb, 2005., str. 164
[2] Manger Robert, Baze podataka, Element, Zagreb, 2012., str. 2
[3] Panian Željko, op. cit., str. 140
[4] loc. cit.

2. Logički model podataka – stvaraju i koriste stručnjaci informatičari (projektanti) kao osnovu za oblikovanje i razvitak informacijskog sustava.
3. Fizički model podataka – predstavlja način na koji stroj (elektroničko računalo) "vidi" podatke pohranjene u njegovoj memoriji.

Prilikom modeliranja podataka u informacijskom sustavu potrebno je ustanoviti entitete, atribute čije će se vrijednosti bilježiti, te odrediti veze između entiteta. Opis entiteta, njihovih atributa i veza između entiteta naziva se konceptualno modeliranje podataka. U skladu s konceptualnim opisom treba odrediti raspored podataka u datotekama odnosno u tablicama. Takav opis podataka naziva se logičko modeliranje podataka. Zadnji korak je odrediti gdje će se podaci nalaziti na fizičkom mediju i kako pristupiti tim podacima. Takav opis naziva se fizički opis podataka.

1.1. Koraci i poslovi pri oblikovanju baze podataka



SLIKA 1: Koraci i poslovi pri oblikovanju baze podataka

Izvor: Čerić Vlatko,Varga Mladen, Informacijska tehnologija u poslovanju, Element, Zagreb, 2004., str. 384

Slika 1 prikazuje korake prilikom izrade baze podataka. Projektiranje podataka obuhvaća konceptualno (popis objekata, atributa i međusobnih odnosa) i logičko (popis datoteka, slogova i polja) modeliranje podataka informacijskog sustava, a obavlja ga administrator. Održavanje baze podataka počinje fizičkim modeliranjem podataka (raspored podataka na mediju), te se nastavlja njezinom fizičkom uspostavom i održavanjem, a obavlja ih administrator baze podataka.

2. KONCEPTUALNO MODELIRANJE PODATAKA

Konceptualno modeliranje podataka odvija se u isto vrijeme kao i prikupljanje informacijskih zahtjeva potrebnih za izgradnju baze podataka. Prilikom prikupljanja informacijskih zahtjeva potrebno je ustanoviti koja će biti uloga tih podataka u budućoj bazi, stoga podaci trebaju biti međusobno povezani, te se moraju međusobno nadopunjavati. Cilj je stvoriti konceptualnu shemu sastavljenu od entiteta, atributa i veza.

Važno svojstvo konceptualne sheme jest da je ona razumljiva ljudima svih struka te da može služiti kao sredstvo za komunikaciju projektanata i korisnika. U toj komunikaciji korisnici nastoje utvrditi jesu li projektanti prepoznali sve poslovne procese, uključili sve potrebne podatke te ispravno shvatili odnose među tim podacima.[5]

Za definiranje podataka i veza među njima koristi se model entiteti-veze.

2.1. Osnovni konstrukti modela entiteti-veze

Nakon prikupljenih svih potrebnih podataka može se početi govoriti o tri klase objekata: entitetima, njihovim atributima, te vezama između entiteta.

2.1.1. Entiteti

Entitet je nešto o čemu se želi spremati podatke, nešto što je u stanju postojati ili ne postojati te se može identificirati.[6]

Može se zaključiti da su oni objekti baze, koji dijele iste karakteristike. Primjeri entiteta su: dobavljač, primka, skladište, artikl, izdatnica, mjesto troška i dr.



SLIKA 2. Prikaz entiteta

(izvor: izradio student)

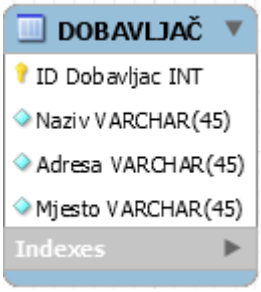
Slika 2 prikazuje primjer entiteta, koji je izrađen u alatu MySQL Workbench.

[5] Manger Robert, op. cit., str. 19
[6] loc. cit.

2.1.2 Atributi

Atributi služe za detaljan opis entiteta. O entitetima se prikupljaju informacije u vidu različitih karakteristika, kao na primjer naziv, adresa i mjesto za entitet dobavljač. U konceptualnom modeliranju takve se karakteristike nazivaju atributima i njihova vrijednost opisuje instancu entiteta.

Atributi mogu biti jednostavni i složeni. Jednostavni atributi se ne mogu podijeliti na više manjih atributa kao što je npr. ID Dobavljač, dok se složeni atributi mogu podijeliti na više jednostavnijih atributa. Primjer složenog atributa je Adresa koji se može podijeliti na ulicu i broj. Kada se govori o atributima može se govoriti o ključnim i neključnim atributima. Razlika između ključnih i neključnih atributa vidljiva je na slici 3, gdje se vidi da je ključni atribut ID Dobavljač koji je ujedno i primarni ključ za razliku od ostalih atributa koji su neključni.



SLIKA 3. Prikaz ključnih i neključnih atributa

(izvor: izradio student)

Slika 3 prikazuje atribute za entitet dobavljač koji se dijele na jednostavne i složene, te na ključne i neključne.

2.1.3. Veze

Veze predstavljaju asocijacije između entiteta. Povezivost označava brojčanu ovisnost između entiteta, odnosno koliko se puta entitet s jedne strane veze pojavljuje u entitetu s druge strane veze. Dakle, uvijek se gleda koliko jedna instanca entiteta, tj. jedan redak u tablici ima veze s drugim entitetom.

Načini na koji veza može povezivati primjerke entiteta određeni su svojstvima funkcionalnosti, obaveznosti članstva, odnosno kardinalnosti. Funkcionalnost je svojstvo koje

kaže može li se veza interpretirati kao preslikavanje (funkcija) iz skupa primjeraka entiteta jednog tipa u skup primjeraka entiteta drugog tipa.[7]

Važno je poznavanje svojstava funkcionalnosti i kardinalnosti da se veza može u idućoj fazi oblikovanja prikazati unutar relacijske sheme. U nastavku su definirana svojstva za binarnu vezu. Važno je napomenuti kako poseban slučaj binarne veze jest veza stanovitog entiteta s tim istim entitetom, takva se veza naziva unarna veza. Binarna i unarna veza smatraju se jednostavnim vezama i prikazuju se binarnim relacijama.

Vrste funkcionalnosti za vezu između tipova entiteta E_1 i E_2: [8]

1. veza jedan naprama jedan (1:1) govori da jedan primjerak prvog tipa entiteta može biti povezan najviše s jednim primjerkom drugog tipa entiteta, također jedan primjerak drugog tipa entiteta može biti povezan najviše s jednim primjerkom prvog tipa entiteta.

2. veza jedan naprama više (1:M) govori da jedan primjerak prvog tipa entiteta može biti povezan s više primjeraka drugog tipa entiteta, također jedan primjerak drugog tipa entiteta može biti povezan samo s jednim primjerkom prvog tipa entiteta.

3. veza više naprama jedan (M:1) govori da jedan primjerak prvog tipa entiteta može biti povezan samo s jednim primjerkom drugog tipa entiteta, također jedan primjerak drugog tipa entiteta može biti povezan s više primjeraka prvog tipa entiteta.

4. veza više naprama više (M:M) govori da jedan primjerak prvog tipa entiteta može biti povezan s više primjeraka drugog tipa entiteta, također da jedan primjerak drugog tipa entiteta može biti povezan s više primjerka prvog tipa entiteta.

Svojstva funkcionalnosti i obaveznosti članstva mogu se otprilike izraziti samo jednim svojstvom koje se zove kardinalnost.[9]

Kada se govori o kardinalnosti moraju se utvrditi dvije kardinalnosti za svaku vezu, za jedan i za drugi smjer. Oznake koje se koriste kod kardinalnosti jesu 0, 1, M te se one odvajaju zarezom. U nastavku su objašnjene navedene oznake za kardinalnost.

[7] ibidem, str. 22
[8] ibidem, str. 23
[9] ibidem, str. 24

Kardinalnost veze promatrane u smjeru od tipa E_1 do tipa E_2:[10]

1. veza 0,1 govori da jedan primjerak prvog tipa entiteta nije povezan ni s jednim ili s najviše jednim primjerkom drugog tipa entiteta.

2. veza 1,1 govori da jedan primjerak prvog tipa entiteta mora biti povezan s točno jednim primjerkom drugog tipa entiteta.

3. veza 0,M govori da jedan primjerak prvog tipa entiteta nije povezan ni s jednim ili s više od jednim primjerkom drugog tipa entiteta.

4. veza 1,M govori da jedan primjerak prvog tipa entiteta mora biti povezan s jednim ili više primjeraka drugog tipa entiteta.

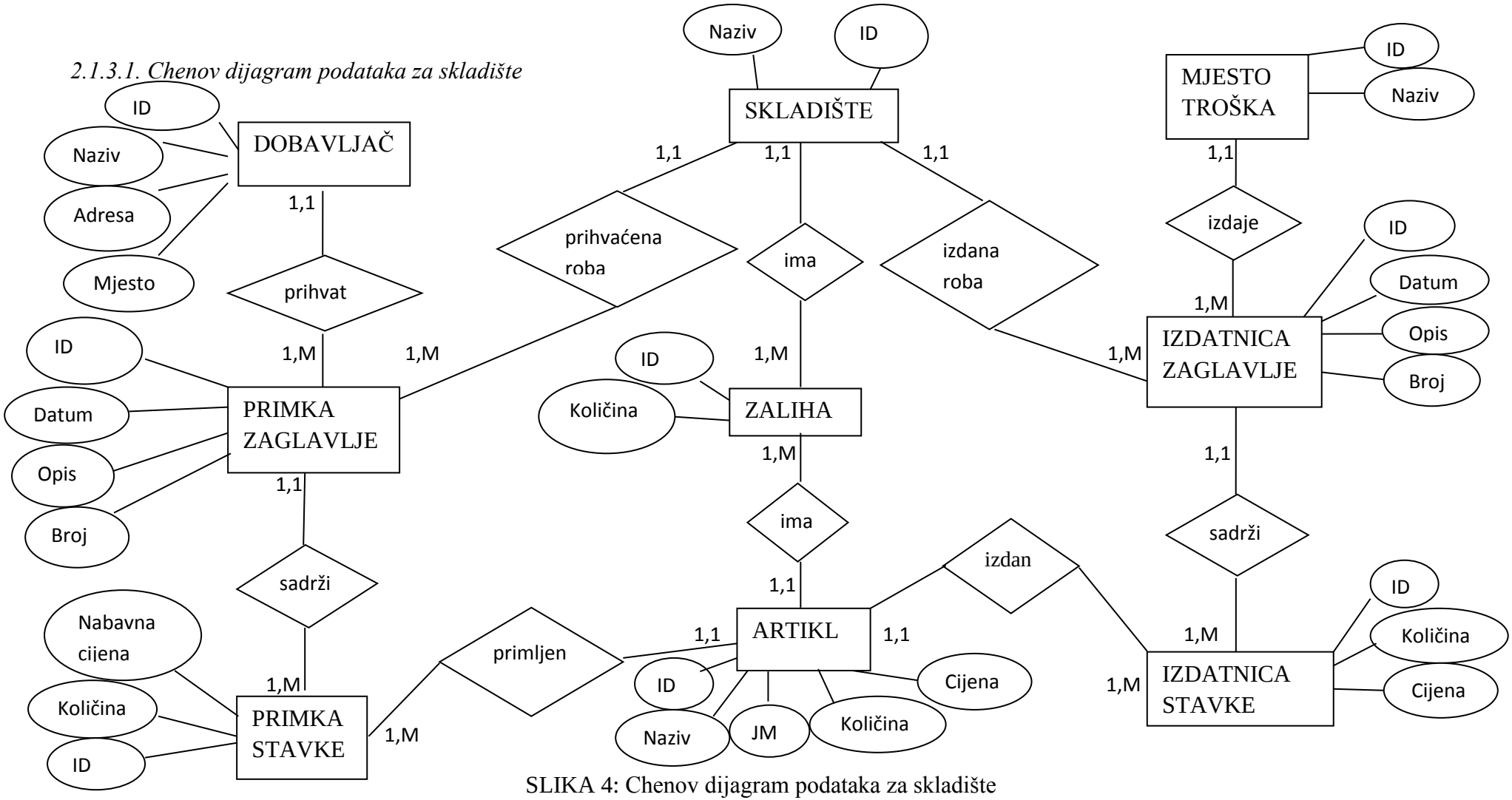
Binarne se veze ubrajaju u jednostavnije veze, te je ta vrsta odnosa između entiteta najpoželjnija zbog svoje jednostavnosti. Postoje situacije kada binarne veze nisu dovoljne, tada treba uvesti složenije veze među koje spada ternarna veza.

Ternarna veza uspostavlja se između tri tipa entiteta. Stanje ternarne veze opisuje se kao skup uređenih trojki primjeraka entiteta koji su trenutačno povezani. Svojstva ternarne veze teže je opisati nego kod binarne veze jer postoji više vrsta funkcionalnosti te više kombinacija za obaveznost članstva. Za istu vezu mogu se promatrati tri kardinalnosti, tako da se na jedan od tri načina odaberu dva od tri tipa entiteta, fiksiraju se primjerci entiteta odabranih dvaju tipova te se gleda broj primjeraka trećeg tipa koji su u vezi s dva fiksirana primjerka.[11]

U nastavku rada prikazani su Chenov, Martinov i Barkerov dijagram podataka za skladište.

[10] ibidem, str. 25
[11] ibidem, str. 33

2.1.3.1. Chenov dijagram podataka za skladište

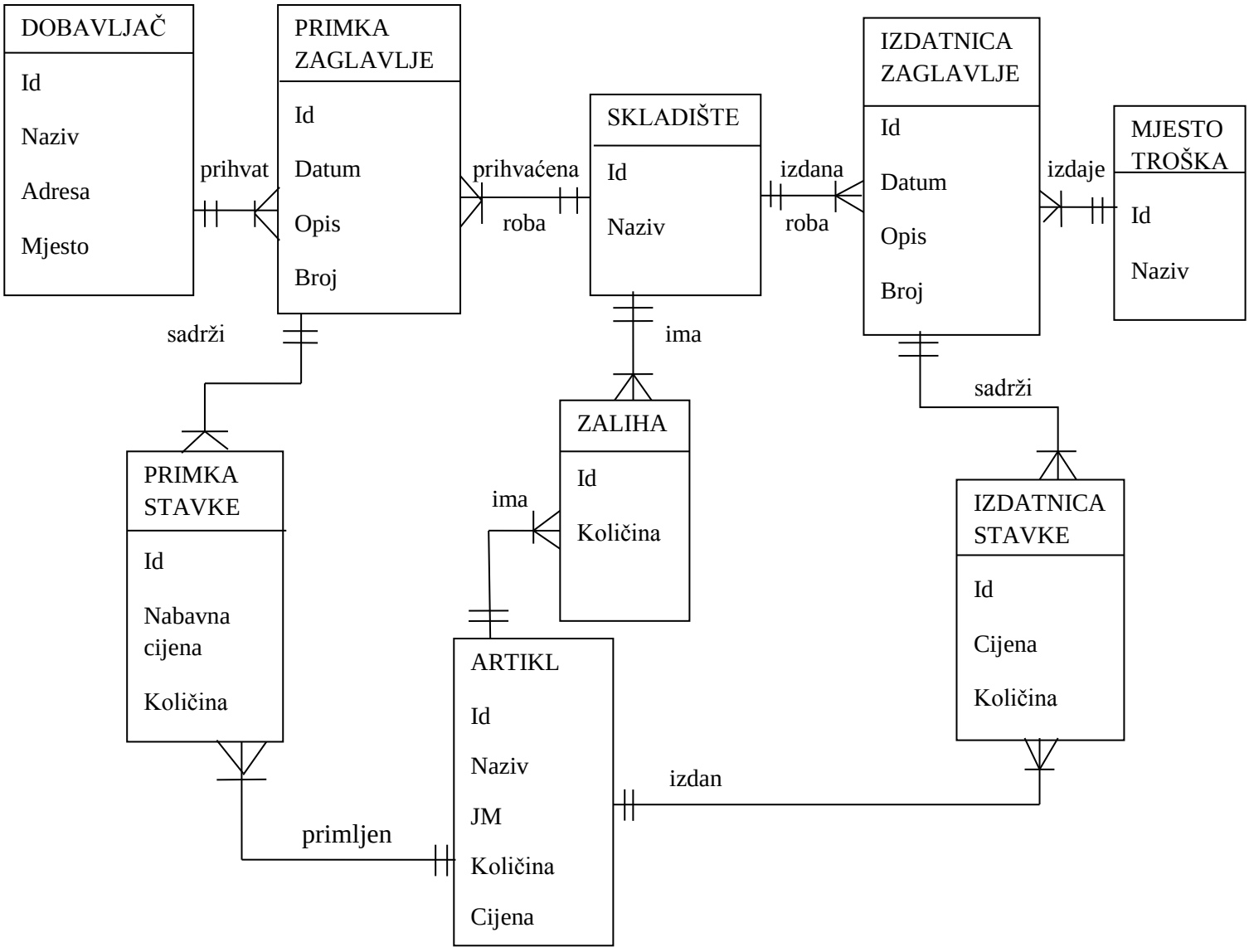


SLIKA 4: Chenov dijagram podataka za skladište

(izvor: izradio student)

Slika 4 prikazuje Chenov dijagram podataka za skladište. Tipovi entiteta su nacrtani kao pravokutnici, a veze kao rombovi. Atributi se nalaze u krugovima koji su spojeni spojnicama s odgovarajućim entitetima. Imena tipova entiteta, odnosno veza upisana su u odgovarajuće pravokutnike, odnosno rombove. Kako bi se znalo između kojih je tipova entiteta uspostavljena veza, odgovarajući romb povezan je spojnicama s odgovarajućim pravokutnicima. Na spojnicama su napisane oznake kardinalnosti veze, a odnose se na onaj entitet pored kojeg su napisane.

2.1.3.2. Martinov dijagram podataka za skladište

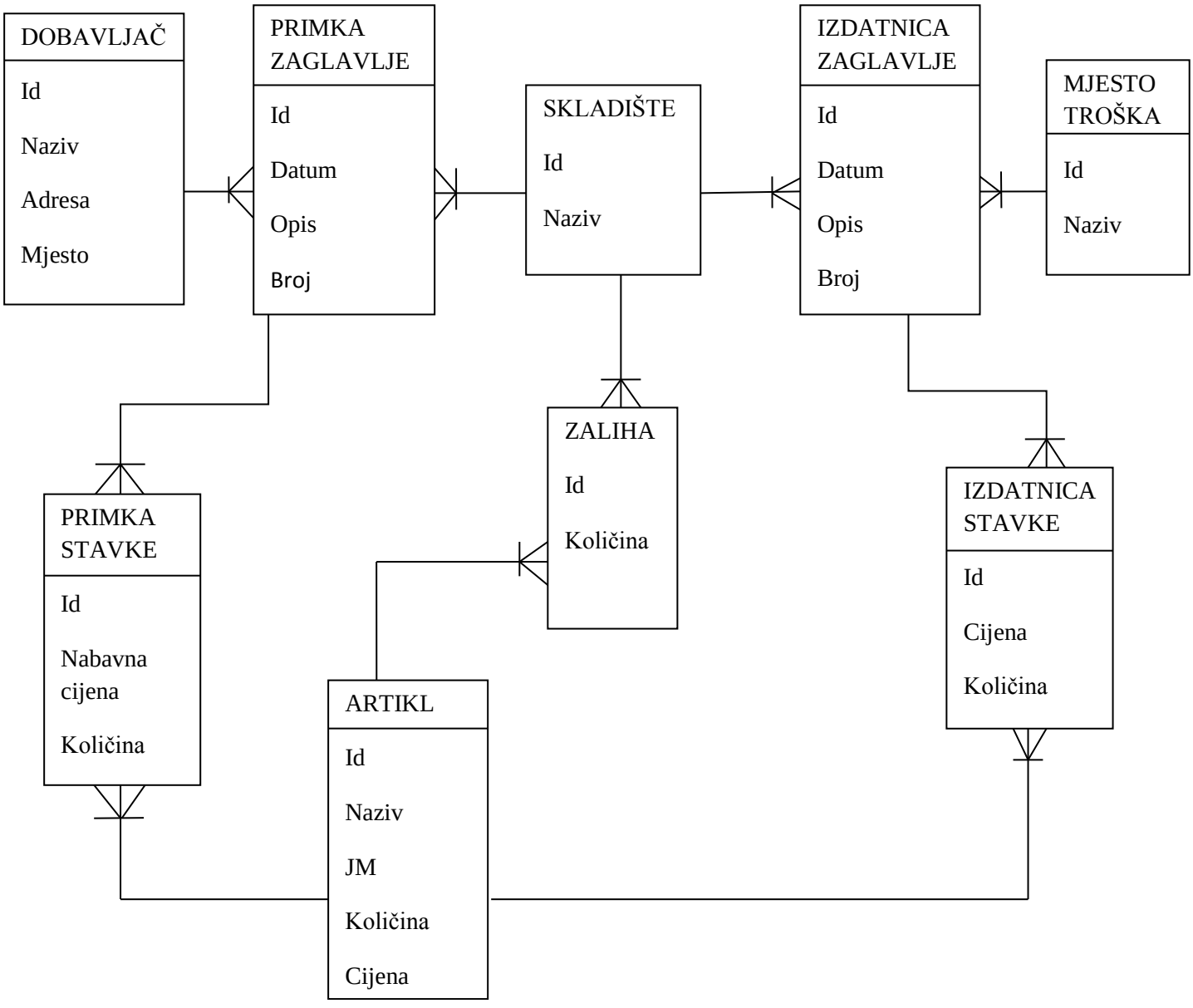


SLIKA 5: Martinov dijagram podataka za skladište

(izvor: izradio student)

Slika 5 prikazuje Martinov dijagram podataka za skladište. Tipovi entiteta i njihovi atributi su nacrtani u pravokutnicima, gdje su atributi od entiteta odvojeni punom linijom. Spojnicama se povezuju različiti tipovi entiteta kako bi se znalo između kojih tipova entiteta postoji određena veza koja je opisana iznad spojnice. Na krajevima spojnica su različite oznake kardinalnosti koje se odnose na onaj tip entiteta pored kojeg se nalaze.

2.1.3.3. Barkerov dijagram podataka za skladište



SLIKA 6: Barkerov dijagram podataka za skladište

(izvor: izradio student)

Slika 6 prikazuje Barkerov dijagram podataka za skladište. Tipovi entiteta i njihovi atributi su nacrtani u pravokutnicima, gdje su atributi od entiteta odvojeni punom linijom. Spojnicama se povezuju različiti tipovi entiteta kako bi se znalo između kojih tipova entiteta postoji određena veza. Na krajevima spojnica su različite oznake kardinalnosti koje se odnose na onaj tip entiteta pored kojeg se nalaze.

2.2. Napredni konstrukti modela entiteti-veze

Prilikom modeliranja podataka ponekad nisu dovoljne tri osnovne klase, te se treba uključiti generalizacija i agregacija.

2.2.1. Generalizacija

Generalizacije je vrsta apstrakcije kod koje se tipovi entiteta niže razine uopćuju tipom entiteta više razine.[12]

Za primjer se mogu uzeti tipovi entiteta primka zaglavlje i primka stavke koji se generaliziraju u tip primka. Takav tip entiteta se zove nadtip ili entitet više razine, dok se tipovi entiteta primka zaglavlje i primka stavke zovu podtip ili entitet niže razine. Generalizacija se opisuje vezom jest. Suprotno od generalizacije je specijalizacija, dakle generalizacija je proces koji ide od nižih razina prema višim, a specijalizacija je proces koji ide od viših razina prema nižim.

2.2.2. Agregacija

Agregacija jest formiranje novog pojma, višeg stupnja, na temelju odnosa postojećih pojmova.[13]

Razlika između generalizacije i agregacije je i u atributima. Dakle, u generalizaciji entitet niže razine proširuje entitet više razine dodatnim atributima što kod agregacije to nije slučaj, kod agregacije entiteti i više i niže razine imaju svoje atribute koji ne trebaju biti ni isti niti međusobno povezani.

[12] Varga Mladen, Baze podataka, konceptualno, logičko i fizičko modeliranje, DRIP, Zagreb 1994., str. 45
[13] loc. cit.

2.3. Izrada modela entiteti-veze

Nakon identifikacije svih važnih elemenata potrebnih za izradu modela entiteti-veze potrebno je napraviti model entiteti-veze za skladište.

Postupak prilikom izrade modela entiteti-veze nije strogo propisan, te općenito treba sadržavati sljedeće:

- Prikupiti i analizirati informacije za izradu modela entiteti-veze
- Izraditi model entiteti-veze gdje treba utvrdit:
  - Entitete
  - Atribute
  - Veze
  - Ključeve
- Konsolidirati model entiteti-veze

Modeliranje podataka započinje prikupljanjem informacija, odnosno poslovnih pravila koje se obavlja intervjuiranjem korisnika, istraživanjem dokumentacije poslovnog sustava i sličnim postupcima. Kod većih informacijskih sustava prikupljanje informacija i izrada modela entiteti-veze može se obaviti po pojedinim problemskim područjima. Nakon intervjuiranja i proučavanja dokumentacije poslovnog sustava analitičar priprema inicijalni poslovni model, koji se sastoji od modela podataka i modela procesa. On ih razvija istovremeno i iterativno.[14]

Nakon prikupljenih i analiziranih informacijskih zahtjeva izrađuje se model entiteti-veze. Prilikom izrade modela entiteti-veze moraju se utvrditi entiteti, atributi i veze između entiteta, te je potrebno utvrditi koji atributi opisuju pojedini entitet. Postoji mogućnost da neki atributi pripadaju i vezi između entiteta. Što se tiče veza mora se za svaku vezu odrediti funkcionalnost i kardinalnost. Entitet mora sadržavati primarni ključ kako bi se identificirala svaka njegova pojava, te je stoga potrebno odabrati primarni ključ za svaki entitet. Prilikom otkrivanja entiteta, atributa i veza može doći do zastoja ako je specifikacija nepotpuna ili nejasna, u tom slučaju projektant treba dodatno porazgovarati s korisnicima da bi se otklonile nejasnoće.

NAPOMENA: Konceptualni model (entiteti-veze) skladišta vidljiv je na slici 17

[14] ibidem, str. 64.

3. LOGIČKO MODELIRANJE PODATAKA

Logičko ili implementacijsko modeliranje polazi od prethodno sastavljenog konceptualnog modela podataka i zahtjeva za korištenje podataka, a rezultira izrađenim logičkim ili implementacijskim modelom. Zahtjevi za korištenjem podataka, opisani kroz model procesa, služe za optimalnu transformaciju konceptualnog u radni logički ili implementacijski model. Ovaj model ne rezultira razradom konačne fizičke strukture podataka pa se naziva logičkim. Budući da se opisuje po pravilima određenog sustava za upravljanje bazom podataka (najčešće relacijskog, objektnog, mrežnog ili hijerarhijskog), model se naziva implementacijski. On se opisuje u obliku sheme baze podataka, u kojoj se koriste pojmovi datoteka ili relacija, logički slog, redak (n-torka), polje ili stupac, primarni ključ, strani ključ itd.[15]

Prvi korak u logičkom modeliranju pretvorba je postojećeg konceptualnog modela (entiteti-veze) u tablice za bazu podataka. Dakle, potrebno je prevesti konceptualni model u logički model. Za opis logičkog modela u ovom radu korišten je relacijski model podataka.

3.1. Relacijski model podataka

U današnje se vrijeme najviše koristi relacijski model podataka zbog svoje jednostavnosti u razumijevanju i praktičnom djelovanju. Teorijski ga je zasnovao Edgar Codd 60-ih godina prošlog stoljeća. Prve realizacije modela bile su jako spore i neefikasne, ali razvojem i napretkom samih računala i informacijske tehnologije dovelo je do napretka i poboljšanja samog modela. Relacijski model je 80-ih godina postao prevladavajući, te se stoga i danas koristi u većini sustava za upravljanje bazom podataka.

U relacijskom modelu baza podataka je sačinjena od tablica tzv. relacija. Relacije se sastoje od više stupaca, te svaka relacija ima svoje ime po kojem se razlikuje od ostalih u istoj bazi podataka. Jedan stupac relacije sadrži vrijednost jednog atributa. Svaki atribut u relaciji ima svoje ime po kojem se razlikuje od ostalih. Relacije se međusobno povezuju preko stupaca koji se nalaze unutar tablica, tj. povezuju se preko stupaca koji se u glavnoj tablici naziva primarni ključ, a u sporednoj tablici strani ključ.

[15] ibidem, str. 91

Osnovni ciljevi relacijskog modela jesu[16]:

- Omogućiti nezavisnost podataka
- Dati teorijske temelje za konzistentno konceptualno postupanje s podacima i za rješenje redundancije podataka
- Omogućiti razvoj skupovno orijentiranih jezika za obradu podataka
- Dati bogat model podataka za opis i obradu jednostavnih i kompleksnih podataka

Relacijski model ostvaruje se jednostavnim tabličnim prikazom podataka i razrađenom teorijom normalizacije podataka. U današnje se vrijeme za rad s relacijskim bazama podataka najviše koristi jezik SQL (Structured Query Language). Od ostalih koji se koriste mogu se spomenuti QUEL i QBE.

PRIMARNI KLJUČ

DOBAVLJAČ

ID Dobavljač	Naziv	Adresa	Mjesto
01	Grga	Valica 9	Pula
02	Velpro	Labinska 2	Pula
03	Roto	Valturska 48	Pula

PRIMARNI KLJUČ

STRANI KLJUČ

PRIMKA ZAGLAVLJE

ID	Datum	Broj	Opis	ID Dobavljač
04	23.4.2015.	10	Dostavljena roba	01
05	05.05.2015.	45	Dostavljena roba	02
06	10.08.2015.	65	Dostavljena roba	03

SLIKA 7 Prikaz relacijskog modela podataka

(izvor: izradio student)

[16] ibidem, str. 92

Slika 7 prikazuje relacijski model podataka koji se sastoji od dvije relacije: Dobavljač i Primka zaglavlje. Relacija Dobavljač obuhvaća sve važne podatke vezano za dobavljača kao što je id, naziv, adresa i mjesto, te relacija Primka zaglavlje također obuhvaća sve važne informacije vezane za primku zaglavlja id, datum, broj i opis. Relacije su povezane primarnim i stranim ključem kako bi se prilikom postavljanja upita mogli dobiti svi važni podaci iz obje tablice.

3.1.1. Struktura relacijskog modela podataka

Strukturu relacijskog modela podataka sačinjavaju: domena, relacija, atribut, n-torke (elementi relacije) i ključevi.

Relacijski model je osnovne koncepte preuzeo iz matematičke teorije skupova, a to su:[17]

Domena – imenovani skup svih vrijednosti koje atribut može poprimiti. Pomoću domene definira se skup vrijednosti koji služe pri modeliranju podataka.

Relacija – imenovani podskup Kartezijevog produkta domene $D_1, D_{2,..,}D_n$: $X(D_1, D_{2,..,}D_n)=$ $\{(d_1, d_2,..,d_n)$: $d_i \in D_i$ za $1 \leq i \leq n\}$ slijedi da je relacija skup n-torki gdje je d_1 vrijednost iz domene D_1, d_2 iz domene D_2,...,d_n iz domene D_n, a n je stupanj relacije.

Relacija r se sastoji od zaglavlja i tijela, gdje je zaglavlje od r zaglavlje n-torke. Relacija r ima iste atribute i isti stupanj kao i zaglavlje. Tijelo od r je skup n-torki koje sve imaju isto zaglavlje, kardinalnost n-torki je i kardinalnost od r. Zaglavlje se još naziva i relacijska shema, a tijelo relacije se još naziva stanje.[18]

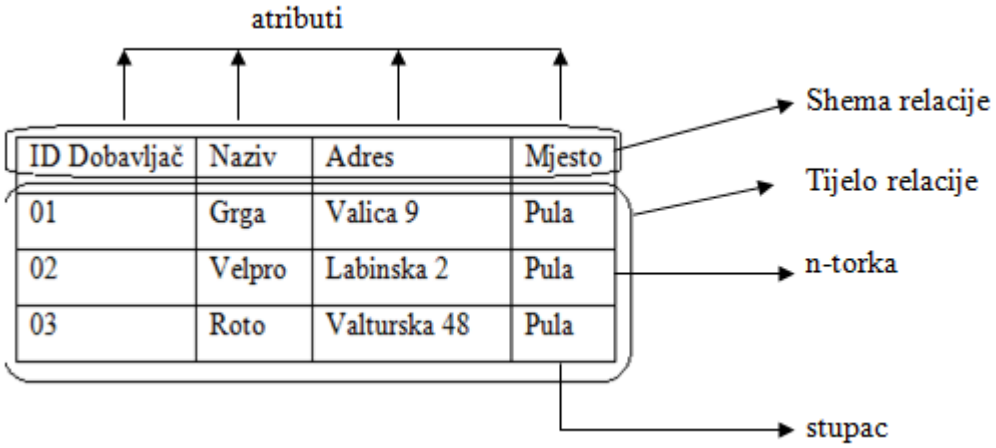
Atribut – imenovani stupac relacije. Atribut može poprimiti jednu vrijednost iz pripadne domene, te se naziv domene može razlikovati od naziva atributa.

Atribut ima svoje ime po kojemu se razlikuje od ostalih u istoj relaciji. Dopušta se da dvije relacije imaju atribute s istim imenom, no tada se podrazumijeva da su to ustvari atributi s istim značenjem. Vrijednosti jednog atributa su podaci iste vrste. Dakle, definiran je tip ili skup dopuštenih vrijednosti za atribut koji se zove domena atributa. Vrijednost atributa mora biti jednostruka i jednostavna (ne ponavlja se, ne da se rastaviti na dijelove). Pod nekim uvjetima tolerira se situacija da vrijednost atributa nedostaje (nije upisana). Jedan redak

[17] ibidem, str. 93
[18] http://e-ucenje.oet.unipu.hr/file.php/30/BP2012_4.pdf

relacije obično predstavlja jedan primjerak entiteta ili bilježi vezu između dva ili više primjerka. Redak se naziva n-torka. U jednoj relaciji ne smiju postojati dvije jednake n-torke. Naime relacija se tumači kao skup n-torki.[19]

Svaka n-torka sadrži samo jednu vrijednost za svaki svoj atribut. Za relaciju koja zadovoljava ovu osobinu kaže se da je normalizirana i da se nalazi u prvoj normalnoj formi.[20]



SLIKA 8 Prikaz relacije

(izvor: izradio student)

Slika 8 prikazuje relaciju koja se sastoji od atributa, sheme, tijela, n-torki i stupaca. U shemi relacije nalaze se svi atributi koji opisuju pojedini entitet. Tijelo relacije sačinjavaju nazivi atributa. N-torka čini jedan redak u relaciji, dok se u stupcima nalaze svi nazivi za svaki atribut posebno.

Ključ je skup atributa čije vrijednosti omogućuju jednoznačno identificiranje svake pojave entiteta u tipu entiteta. Entitet može imati jedan ili više ključeva, a mora imati barem jedan ključ. Uvijek se nastoji pronaći minimalni ključ, čiji podskup atributa (naravno, ako se ključ sastoji od više atributa) više nema karakteristike ključa. Minimalnih ključeva može biti više. Oni se nazivaju mogućim ili kandidatnim ključevima (eng. candidate key). Jedan izabrani mogući ključ postaje primarni ključ(eng. primary key). Nakon izbora primarnog ključa ostali mogući ključevi postaju alternativni ključevi (eng. alternate key). Svaki entitet mora imati primarni ključ kojim se jednoznačno identificira svaka njegova pojava. Za primarni ključ se

[19] Manger Robert, op. cit., str. 38
[20] http://e-ucenje.oet.unipu.hr/file.php/30/BP2012_4.pdf

često koristi automatski generirani brojač koji ne nosi podatke jer je broj bolji izbor nego niz znakova promjenjive duljine.[21]

Kandidat za primarni ključ mora zadovoljiti dva uvjeta:[22]

1. Uvjet jednoznačnosti koji govori da u skupu entiteta ne smiju postojati dvije n-torke s istim vrijednostima svih ključnih atributa.

2. Uvjet minimalnosti koji govori da ne postoji podskup atributa ključa koji također zadovoljava uvjet jednoznačnosti.

Strani ključ (eng. foreign key) je skup atributa koji se odnosi na ključ drugog tipa entiteta, tj. skup atributa čije se vrijednosti odnose na vrijednost ključa drugog entiteta. Strani ključ može poprimiti vrijednost primarnog ključa drugog entiteta ili može poprimiti NULL vrijednost, ukazuje na povezanost između entiteta, odnosno skupova entiteta.[23]

3.1.1.1. Pregled tablica

Tablica 1: Dobavljač

'Id Dobavljač', INT, NOT NULL;

'Naziv', VARCHAR (45), NOT NULL;

'Adresa', VARCHAR (45), NOT NULL;

'Mjesto', VARCHAR (45), NOT NULL;

Tablica 2: Primka zaglavlje

'Id Primka Zaglavlje', INT, NOT NULL;

'Datum', DATE, NOT NULL;

'Opis', VARCHAR (100), NOT NULL;

'Broj', INT, NOT NULL;

'Id Dobavljač', INT, NOT NULL;

'Id Skladišta', INT, NOT NULL;

[21] https://www.fkit.unizg.hr/_download/repository/Pred_BAZE_PR-2012-13_MV_konacno.pdf
[22] https://www.fkit.unizg.hr/_download/repository/Pred_BAZE_PR-2012-13_MV_konacno.pdf
[23] https://www.fkit.unizg.hr/_download/repository/Pred_BAZE_PR-2012-13_MV_konacno.pdf

Tablica 3: Primka stavke

'Id Primka stavke', INT, NOT NULL;

'Količina', INT, NOT NULL;

'Nabavna cijena', INT, NOT NULL;

'Id Primka zaglavlje', INT, NOT NULL;

'Id Artikl', INT, NOT NULL;

Tablica 4: Skladište

'Id Skladišta', INT, NOT NULL;

'Naziv skladišta', VARCHAR(45) NOT NUL;

Tablica 5: Zaliha

'Id Zaliha', INT, NOT NULL;

'Količina', INT, NOT NULL;

'Id Skladišta', INT, NOT NULL;

'Id Artikl', INT, NOT NULL;

Tablica 6: Artikl

'Id Artikl', INT, NOT NULL;

'Naziv artikla', VARCHAR(45), NOT NULL;

'JM', VARCHAR(45), NOT NULL;

'Količina', INT, NOT NULL;

'Cijena', INT, NOT NULL;

Tablica 7: Izdatnica zaglavlje

'Id Izdatnica zaglavlje', INT, NOT NULL;

'Datum', DATE, NOT NULL;

'Opis', VARCHAR(100), NOT NULL;

'Broj', INT, NOT NULL;

'Id Skladišta', INT, NOT NULL;

'Id Mjesto troška', INT, NOT NULL;

Tablica 8: Izdatnica stavke

'Id Izdatnica stavke', INT, NOT NULL;

'Količina', INT, NOT NULL;

'Cijena', INT, NOT NULL;

'Id Izdatnica zaglavlje', INT, NOT NULL;

'Id Artikl', INT, NOT NULL;

Tablica 9: Mjesto troška

'Id Mjesto troška', INT, NOT NULL;

'Naziv', VARCHAR(45), NOT NULL;

Opisane tablice su prikazane na slici 18

3.1.2. Ograničenja

U relacijskom modelu postoje dva opća ograničenja: [24]

1. Entitetni integritet – je kombinacija ograničenja jedinstvenosti i ne-nulvrijednosti. Definira ograničenje ključa i jamči jednoznačno identificiranje bilo koje n-torke u bazi podataka.

2. Referencijalni integritet – ima veliku važnost za konzistentnost baze podataka. Povezuje pojam primarnog ključa sa stranim ključem. Ako u relaciji R postoji strani ključ koji odgovara primarnom ključu relacije S, svaka vrijednost stranog ključa u relaciji R mora biti jednaka vrijednosti primarnog ključa u nekoj n-torki relacije S ili jednaka nul-vrijednosti (R i S mogu biti ista relacija).

[24] https://www.fkit.unizg.hr/_download/repository/Pred_BAZE_PR-2012-13_MV_konacno.pdf

3.1.3. Operacije

U relacijskom modelu koriste se i različite operacije koje služe za uspješno modeliranje određene baze podataka. Postoji osam različitih operacija koje se dijele na jednostavne i složene. Jednostavne operacije sačinjavaju: unija, presjek, razlika i Kartezijev produkt, dok složene operacije sačinjavaju: selekcija, projekcija, spajanje i dijeljenje. U nastavku rada objašnjena je svaka operacija posebno.

3.1.3.1. Unija

Unija dviju kompatibilnih relacija R(A_1, A_2, A_3,..., A_n) i S(A_1, A_2, A_3,..., A_n) nova je relacija T(A_1, A_2, A_3,..., A_n) koja se sastoji od svih n-torki sadržanih u R i S.[25] Oznaka za uniju je ∪ što se označava R ∪ S.

ARTIKL 1

ID	NAZIV
1	Čips
2	Cola
3	Pizza

ARTIKL 2

ID	NAZIV
2	Cola
4	Smoki

ARTIKL 1∪ ARTIKL 2

ID	NAZIV
1	Čips
2	Cola
3	Pizza
4	Smoki

SLIKA 9 Operacija unije relacija ARTIKL 1 i ARTIKL 2

(izvor: izradio student)

Slika 9 prikazuje operaciju unije relacija Artikl 1 i Artikl 2, gdje se dobiva nova relacija Artikl 1 ∪ Artikl 2 koja se sastoji od svih n-torki sadržanih u relacijama Artikl 1 i Artikl 2.

3.1.3.2. Presjek

Presjek dviju unijski kompatibilnih relacija R(A_1, A_2, A_3,..., A_n) i S(A_1, A_2, A_3,..., A_n) jest nova relacija T(A_1, A_2, A_3,..., A_n) koja obuhvaća sve n-torke što se istovremeno nalaze u relaciji R i u relaciji S.[26] Oznaka za presjek je ∩, što se označava R ∩ S.

[25] Varga Mladen, op. cit., str. 96
[26] ibidem, str. 101

ARTIKL 1

ID	NAZIV
1	Čips
2	Cola
3	Pizza
4	Smoki
5	Čokolada

ARTIKL 2

ID	NAZIV
2	Cola
4	Smoki
9	Bomboni

ARTIKL 1∩ ARTIKL 2

ID	NAZIV
2	Cola
4	Smoki

SLIKA 10 Operacija presjeka relacija ARTIKL 1 i ARTIKL 2

(izvor: izradio student)

Slika 10 prikazuje operaciju presjeka relacija Artikl 1 i Artikl 2, gdje se dobiva nova relacija Artikl 1 ∩ Artikl 2 koja se sastoji od n-torki koje se istovremeno nalaze u relacijama Artikl 1 i Artikl 2.

3.1.3.3. Razlika

Razlika odnosno diferencijacija dviju unijski kompatibilnih relacija R(A_1, A_2, A_3,..., A_n) i S(A_1, A_2, A_3,..., A_n) jest nova relacija T(A_1, A_2, A_3,..., A_n) koja obuhvaća sve n-torke relacije R koje istovremeno nisu sadržane u relaciji S.[27] Oznaka za razliku je \, što se označava R \ S.

ARTIKL 1

ID	NAZIV
1	Čips
3	Pizza
5	Čokolada

ARTIKL 2

ID	NAZIV
3	Pizza
4	Smoki

ARTIKL 2\ARTIKL 1

ID	NAZIV
4	Smoki

ARTIKL 1\ARTIKL 2

ID	NAZIV
1	Čips
5	Čokolada

SLIKA 11 Operacija razlike relacija ARTIKL 1 i ARTIKL 2

(izvor: izradio student)

Slika 11 prikazuje operaciju razlike relacija Artikl 1 i Artikl 2, gdje se dobivaju dvije nove relacije Artikl 2 \ Artikl 1 i Artikl 1 \ Artikl 2. Relacija Artikl 2 \ Artikl 1 sastoji se od n-torki relacije Artikl 2 koje nisu istovremeno sadržane u relaciji Artikl 1, odnosno relacija Artikl 1 \

[27] ibidem, str. 97

Artikl 2 sastoji se od n-torki relacije Artikl 1 koje nisu istovremeno sadržane u relaciji Artikl 2.

3.1.3.4. Kartezijev produkt

Kartezijev produkt dviju relacija R(A_1, A_2, A_3,..., A_n) i S(B_1, B_2, B_3,..., B_m) nova je relacija T(A_1, A_2, A_3,...,A_n, B_1, B_2, B_3,...,B_m) koja se sastoji od n-torki nastalih spajanjem svake n-torke relacije R sa svakom n-torkom relacije S.[28]

Oznaka za Kartezijev produkt je x, što se označava R x S.

DOBAVLJAČ

ID	NAZIV
1	Grga
2	Velpro

ARTIKL

ŠIFRA	NAZIV
01	Čips
02	Cola

DOBAVLJAČxARTIKL

ID	NAZIV	ŠIFRA	NAZIV
1	Grga	01	Čips
1	Grga	02	Cola
2	Velpro	01	Čips
2	Velpro	02	Cola

SLIKA 12 Operacija Kartezijevog produkta relacija DOBAVLJAČ i ARTIKL

(izvor: izradio student)

Slika 12 prikazuje operaciju Kartezijevog produkta relacija Dobavljač i Artikl, gdje se dobiva nova relacija Dobavljač x Artikl koja je nastala spajanjem svake n-torke relacije Dobavljač sa svakom n-torkom relacije Artikl.

3.1.3.5. Selekcija

Selekcija relacije R(A_1, A_2, A_3,..., A_n) jest nova relacija T(A_1, A_2, A_3,..., A_n) koja se sastoji od n-torki relacije R koje ispunjavaju zadani uvjet. Selekcija (restrikcija) jest unarna operacija kojom se iz relacije izdvaja određen skup n-torki koje ispunjavaju zadani uvjet. Uvjet može biti jednostavan ili kompleksan. Jednostavan atribut opisuje se formom <atribut> <operator>

[28] loc. cit.

<vrijednost> gdje je operator jedan od theta-operatora {=, ≠, <, ≤, >, ≥}. U kompleksnim uvjetima koriste se logički operatori {AND, OR, NOT} odnosno {I, ILI, NE}.[29]

Oznaka za selekciju je $\sigma_{uvjet}(R)$.

DOBAVLJAČ

ID	NAZIV	MJESTO
1	Grga	Pula
2	Velpro	Pula
3	Roto	Rijeka

$\sigma_{MJESTO=Pula}$(DOBAVLJAČ)

ID	NAZIV	MJESTO
1	Grga	Pula
2	Velpro	Pula

$\sigma_{MJESTO=Pula\ AND\ ID<2}$(DOBAVLJAČ)

ID	NAZIV	MJESTO
1	Grga	Pula

SLIKA 13 Operacija selekcije relacije DOBAVLJAČ

(izvor: izradio student)

Slika 13 prikazuje operaciju selekcije za relaciju Dobavljač. Da bi se dobilo odgovarajuće rješenje potrebno je zadovoljiti sljedeće uvjete. U primjeru na slici 13 potrebno je zadovoljiti dva uvjeta. Prvi uvjet se odnosi na sve dobavljače koji su iz Pule, a drugi uvjet je da prikaže one dobavljače koji imaju ID < 2. Dakle nova relacija $\sigma_{MJESTO=Pula\ AND\ ID<2}$(DOBAVLJAČ) se sastoji od n-torki relacije Dobavljač koje ispunjavaju zadani uvjet.

3.1.3.6. Projekcija

Projekcija relacije R nova je relacija T koja se sastoji od atributa relacije R po kojima je obavljena operacija projekcije, i u kojoj su uklonjene jednake n-torke. Ako je X podskup atributa relacije R, projekcija relacije R po X je relacija R' dobivena uklanjanjem atributa R-X i uklanjanjem višestrukih n-torki. Projekcija je unarna operacija kojom se iz relacije izdvajaju pojedini stupci (po kojima se obavlja projiciranje relacije). Stupci koji se izdvajaju (projiciraju) navode se kao parametri operacije.[30] Oznaka za projekciju je $\Pi_{Ai, Aj,..., Am}(R)$.

[29] ibidem, str. 98
[30] ibidem, str. 99

DOBAVLJAČ

ID	NAZIV	MJESTO
1	Grga	Pula
2	Velpro	Pula
3	Roto	Rijeka

$\Pi_{ID,NAZIV}$(DOBAVLJAČ)

ID	NAZIV
1	Grga
2	Velpro
3	Roto

Π_{MJESTO}(DOBAVLJAČ)

MJESTO
Pula
Rijeka

SLIKA 14 Operacija projekcije relacije DOBAVLJAČ

(izvor: izradio student)

Slika 14 prikazuje operaciju projekcije za relaciju Dobavljač. Operacija projekcije uklanja iste n-torke koje se nalaze u relaciji Dobavljač, što je i vidljivo na primjeru Π_{MJESTO}(DOBAVLJAČ).

3.1.3.7. Spajanje

Theta-spoj jest selekcija Kartezijevog produkta relacija R i S opisana formulom A<operator>B, gdje je A atribut relacije R, a B atribut relacije S. Theta-spoj (eng. theta-join) skraćeno se naziva spojem. Relacija dobivena spajanjem relacija R i S sadrži n-torke relacije R spojene s n-torkama relacije S, koje udovoljavaju uvjetu A<operator>B. Kao operator koristi se jedan od theta-operatora $\{=, \neq, <, \leq, >, \geq\}$.[31] Oznaka za spajanje je R[uvjet]S.

Kod proširenog theta-spajanja operator je složen od više theta-operatora kombiniranih logičkim operatorima {AND, OR, NOT} kao kod selekcije. Primjena proširenog spajanja je rijetka, ali moguća.[32]

[31] loc. cit.
[32] ibidem, str. 100

PRIMKA STAVKE

ID PS	KOLIČINA PS	NAZIV	NABAVNA CIJENA
PS1	50	Cola	10
PS2	55	Redbull	12
PS3	40	Čokolada	9
PS4	60	Bomboni	5

IZDATNICA STAVKE

ID IS	KOLIČINA IS	NAZIV	IZDANA CIJENA
IS1	40	Cola	20
IS2	55	Redbull	25
IS3	50	Čips	12

PRIMKA STAVKE[NAZIV=NAZIV] IZDATNICA STAVKE

ID PS	KOLIČINA PS	NAZIV	NABAVNA CIJENA	ID IS	KOLIČINA IS	NAZIV	IZDANA CIJENA
PS1	50	Cola	10	IS1	40	Cola	20
PS2	55	Redbull	12	IS2	55	Redbull	25

PRIMKA STAVKE[NAZIV=NAZIV AND KOLIČINA PS=KOLIČINA IS]

ID PS	KOLIČINA PS	NAZIV	NABAVNA CIJENA	ID IS	KOLIČINA IS	NAZIV	IZDANA CIJENA
PS2	55	Redbull	12	IS2	55	Redbull	25

SLIKA 15 Operacija theta-spajanja relacija PRIMKA STAVKE i IZDATNICA STAVKE

(izvor: izradio student)

Slika 15 prikazuje operaciju theta-spajanja relacija Primka stavke i Izdatnica stavke. Na primjeru relacije PRIMKA STAVKE [NAZIV=NAZIV] IZDATNICA STAVKE potrebno je ispuniti uvjet [NAZIV=NAZIV] kako bi se dobila relacija u kojoj se nalaze proizvodi istog naziva iz relacija Primka stavke i Izdatnica stavke. Isto kao i u prethodnom slučaju kako bi se dobila relacija PRIMKA STAVKE [NAZIV=NAZIV AND KOLIČINA PS=KOLIČINA IS] IZDATNICA STAVKE potrebno je ispuniti odgovarajući uvjet, a to je da nazivi i količine iz relacija Primka stavke i Izdatnica stavke budu jednake.

3.1.3.8. Dijeljenje

Dijeljenje relacije R(A_1, A_2, A_3,..., A_D, A_{D+1},..., A_n) s relacijom S(A_{D+1},..., A_n) daje novu relaciju T(A_1, A_2, A_3,..., A_D) tako da je T najveća moguća relacija sa svojstvom da sve n-torke relacije T spojene sa svim n-torkama relacije S daju n-torke sadržane u R. Ako je x skup n-

torki relacije S, dijeljenje je jednostavan način da se pronađe skup n-torki y za koje vrijedi da su <x,y> sadržani u relaciji R.[33]

DOBAVLJAČ

ID	DOSTAVIO
1	Čokolada
2	Čips
2	Bomboni
1	Bomboni
3	Čips
3	Bomboni
2	Cola

UVJET

ARTIKL
Čips
Bomboni

DOBAVLJAČ [DOSTAVIO/ARTIKL] UVJET

ID
2
3

SLIKA 16 Operacija dijeljenja relacija DOBAVLJAČ i UVJET

(izvor: izradio student)

Slika 16 prikazuje operaciju dijeljenja relacija Dobavljač i Uvjet, gdje se dobiva nova relacija DOBAVLJAČ [DOSTAVIO/ARTIKL] UVJET koja nastaje tako da sve n-torke relacije DOBAVLJAČ [DOSTAVIO/ARTIKL] UVJET spojene sa svim n-torkama relacije Uvjet daju n-torke sadržane u relaciji Dobavljač.

3.1.4. Pretvorba konceptualnog modela (entiteti-veze) u logički (relacijski) model

U pravila pretvorbe spadaju:

- Pretvorba entiteta u relacije
- Pretvorba veza
- Pretvorba atributa
- Normalizacija dobivene relacijske sheme

Pretvorba entiteta u relacije:[34]

Svaki entitet tvori jednu relacijsku shemu. Ključ entiteta postaje primarnim ključem relacije. Detaljnije, po tipu entiteta:

[33] ibidem, str. 102
[34] ibidem, str. 145

E1) Jaki entiteti tvore samostalne relacije. Primarni ključ relacije jednak je ključu entiteta.

E2) Identifikacijski slabi entiteti tvore zavisne relacije. Primarni ključ slabog entiteta sastoji se od ključa nadređenog jakog entiteta i diskriminatora slabog entiteta, pa se i primarni ključ relacije slabog entiteta sastoji od dva dijela: primarnog ključa relacije jakog entiteta i dodatnih atributa zavisne relacije.

E3) Entiteti podtipovi tvore podrelaciju. Kod ekskluzivne specijalizacije u relaciju se entiteta nadtipa može uvesti atribut po kojem je načinjena specijalizacija.

Pretvorba veza:[35]

V1) U vezi 1:1 (jedan-prema-jedan) uvesti strani ključ u relaciju s manje n-torki ili zadržati prirodnu vezu "roditelj-dijete". Općenito je u takvoj simetričnoj vezi moguće uvođenje stranog ključa u bilo koju relaciju. Moguća su i druga rješenja, ovisno o kardinalnosti veze:

a) Veza 1,1:1,1 se može riješiti i uvođenjem svih atributa jednog entiteta u relaciju drugog entiteta, jer je broj pojava oba entiteta jednak. Za primarni ključ se izabire ključ jednog od entiteta koji sudjeluju u vezi.
b) U vezi 0,1:1,1 uvodi se strani ključ u relaciju entiteta koji obavezno sudjeluje u vezi.
c) Veza 0,1:0,1 može se riješiti i uvođenjem nove relacije kao u vezi M:M (pravilo V3), ali s jedinom razlikom da primarni ključ nije sastavljen od oba ključa entiteta, nego se za primarni ključ izabire ključ jednog od entiteta.

V2) U vezi 1:M (jedan-prema-više) uvesti strani ključ u relaciju na strani M. Ovo sigurno vrijedi za vezu 1,1:0,M. Za vezu 0,1:0,M drugo rješenje je uvođenje nove relacije kao u vezi M:M (pravilo V3), ali s jedinom razlikom da primarni ključ nije sastavljen od oba ključa entiteta, nego se za primarni ključ uzima ključ entiteta na strani M.

V3) U vezi M:M (više-prema-više) otvoriti novu relaciju. Pravilo vrijedi za sve kombinacije kardinalnosti veze M:M. Nova relacija naziva se agregacijskom, asocijativnom ili veznom. Primarni ključ agregacijske relacije uvijek je sastavljen od primarnih ključeva relacija koje čine agregaciju.

[35] ibidem, str. 146

V4) Ternarne i ostale veze višeg stupnja tvore novu relaciju. Realizacija je slična binarnoj vezi M:M. Nova relacija ima toliko stranih ključeva koliki je stupanj veze. U najčešćoj ternarnoj vezi tri su strana ključa i od njih se formira primarni ključ.

a) 1:1:1 – mogući ključevi: ključ A - ključ B, ključ A - ključ C, ključ B - ključ C
b) 1:1:M – mogući ključevi: ključ A - ključ C, ključ B - ključ C
c) 1:M:M – primarni ključ: ključ B – ključ C
d) M:M:M – primarni ključ: ključ A – ključ B – ključ C

Pretvorba atributa:[36]

A1) Atribut koji ima više vrijednosti (tzv. ponavljajuća grupa) tvori novu zavisnu relaciju.

A2) Obavezne atribute opisati ograničenjem "ne nul-vrijednosti"= NOT NULL.

A3) Atribute s jedinstvenim vrijednostima opisati ograničenjem "jedinstvenosti vrijednosti"= UNIQUE.

A4) Atribute strane ključeve opisati ograničenjem referencijalnog integriteta.

A5) Atribute s ograničenom domenom vrijednosti opisati ograničenjem uz kontrolu odnosno navođenjem domene u kojoj je ograničenje već opisano.

A6) Atributi veze, koja se rješava uvođenjem stranog ključa, idu u relaciju zajedno sa stranim ključem.

A7) Atributi veze, koja se rješava uvođenjem nove agregacijske relacije, postaju atributi nove relacije.

U relacijskom modelu mogu se jasno vidjeti koji su primarni, a koji strani ključevi, te se mogu vidjeti koji atributi opisuju pojedinu tablicu.

Pretvorba konceptualnog modela (entiteti-veze) u logički (relacijski) model vidljiv je na slikama 17 i 18

[36] ibidem, str. 149

3.1.5. Normalizacija

Proces provođenja normalnih formi nad tablicom, odnosno dovođenje tablice u stanje koja određena normalna forma zahtjeva naziva se normalizacijom. Postiže se tako da se analiziraju atributi tablice, te po potrebi razdvajaju na više manjih tablica. Postoji sedam normalnih formi od kojih se prve tri redovito koriste u praksi, a ostale četiri se nazivaju višim normalnim formama te se koriste rjeđe. U ovom radu objašnjene su prve tri normalne forme koje se često koriste u praksi.

3.1.5.1. Prva normalna forma

Svojstvo jednostrukosti i jednostavnosti zove se jednim imenom svojstvo prve normalne forme (oznaka: 1NF). Dakle, kažemo da je relacija u prvoj normalnoj formi jer su sve vrijednosti njezinih atributa jednostruke i nedjeljive, odnosno ako svaki atribut ima jednu vrijednost.[37]

- SKLADIŠTA (ID Skladišta, Naziv Skladišta, ID Zaliha, Količina, ID Artikl, Naziv, JM, Količina, Cijena)

Primjer prve normalne forme koji vidljiv na tablici Skladišta s atributima ID Skladišta, Naziv Skladišta, ID Zaliha, Količina, ID Artikl, Naziv, JM, Količina i Cijena. Svaki atribut ima svoju vrijednost, te se stoga ta tablica već nalazi u prvoj normalnoj formi i nije ju potrebno rastavljati.

3.1.5.2. Druga normalna forma

Relacija je u drugoj normalnoj formi (oznaka: 2NF) ako je svaki njezin neprimarni atribut potpuno funkcionalno ovisan o primarnom ključu. Drugim riječima, relacija je u 2NF ako u njoj nema parcijalnih ovisnosti atributa o primarnom ključu. Relacija s jednostavnim ključem (dakle, ključem koji se sastoji samo od jednog atributa) automatski je u 2NF.[38]

3.1.5.3. Treća normalna forma

Relacija je u trećoj normalnoj formi (oznaka: 3NF) ako je u 2NF i ako ne sadržava tranzitivne ovisnosti. Preciznije, relacija R je u 3NF ako za svaku funkcionalnu ovisnost $X \rightarrow A$ u R, takvu da A nije dio od X, vrijedi: X sadržava ključ za R ili je A primarni atribut.[39]

[37] Manger Robert, op. cit., str. 49
[38] ibidem, str. 52
[39] ibidem, str. 54

SKLADIŠTA (ID Skladišta, Naziv Skladišta, ID Zaliha, Količina, ID Artikl, Naziv, JM, Količina, Cijena)

- SKLADIŠTA (ID Skladišta, Naziv skladišta)
- ZALIHA (ID Zaliha, Količina, ID Skladišta, ID Artikl)
- ARTIKL (ID Artikl, Naziv, JM, Količina, Cijena)

Primjer treće normalne forme vidljiv je na tablici Skladišta s atributima ID Skladišta, Naziv skladišta, Zaliha, Količina, Artikl, JM, Količina i Cijena gdje je ID Skladišta primarni ključ, te vrijede zavisnosti Skladišta - Zaliha i Zaliha – Artikl. Vidljivo je kako vrijedi zavisnost Skladišta - Artikl, stoga je potrebno izvršiti dekompoziciju po zavisnosti Zaliha - Artikl. Dobivene tablice nalaze se u 3NF

3.1.6. SQL

SQL je najraširenija jezik za rad s relacijskom bazom podataka. Sama kratica znači Structured Query Language. Jezik je nastao u 70-im godinama 20. stoljeća u sklopu razvojno-istraživačkog projekta system R unutar kompanije IBM. Jezik se postupno usavršavao, a njegova dotjerana varijanta pojavljuje se u današnjem IBM-ovu relacijskom DBMS-u zvanom DB2. U širenju SQL-a tijekom 80-ih godina važnu ulogu odigrala je softverska kuća Oracle Corporation: ona je ugradila SQL u svoj DBMS, te ga je time učinila dostupnim i popularnim na svim važnijim računalnim platformama. Uporabu SQL-a u razvoju web aplikacija tijekom 90-ih godina potakla je kompanija MySQL AB svojim besplatnim DBMS-om. U skladu sa svojim imenom, SQL u prvom redu omogućuje postavljanje upita u relacijskim bazama podataka.[40]

Za rad s relacijskom bazom podataka koriste se tzv. upitni jezici. Oni su neproceduralni, za razliku od proceduralnih operacija relacijske algebre. Prednost je u tome što se neproceduralnim jezicima ne opisuje postupak (procedura) rješenja, nego uvjeti rješenja. Postoji više upitnih jezika za rad s relacijskom bazom podataka. Najviše se koristi SQL, naročito otkako je postao standardnim jezikom za rad s relacijskom bazom podataka. Iako se naziva upitnim jezikom, SQL ima naredbe za sveobuhvatan rad s relacijskom bazom podataka. To su naredbe za definiranje baze podataka (DDL-naredbe, eng. Data Definition

[40] ibidem, str. 84

Language), naredbe za manipulaciju podacima (DML-naredbe, eng. Data Manipulation Language) i upravljačke naredbe.[41]

Postoje četiri osnovne DML naredbe za upravljanje podacima:

- SELECT (postavljanje SQL upita nad bazom podataka)
- INSERT (unos n-torke u relaciju)
- DELETE (brisanje n-torke iz relacije)
- UPDATE (ažuriranje n-torke u relaciji)

Prilikom postavljanja upita u ovom završnom radu rabila se naredba SELECT koja daje popis polja o kojima se žele prikupljati podaci, te sljedeći uvjeti (FROM, WHERE, ORDER BY) koji su morali biti ispunjeni kako bi se dobili željeni podaci. FROM navodi tablice koje sadrže polja navedena u naredbi SELECT, WHERE određuje kriterije polja s kojima se svaki zapis mora podudarati da bi bio uvršten u rezultate, te ORDER BY opisuje način sortiranja podataka ovisno o tome da li pored ORDER BY stoji DESC ili ASC, što znači padajuće sortiranje podataka odnosno rastuće sortiranje podataka.

No jezik se ne zaustavlja na tome: postoje i naredbe za stvaranje relacije, unos, promjenu i mijenjanje podataka, upravljanje transakcijama, davanje i oduzimanje ovlaštenja korisnicima. Upitni dio SQL-a uglavnom je zasnovan na relacijskom računu, s time da je matematička notacija zamijenjena ključnim riječima nalik na govorni engleski jezik. No lagano se realiziraju i sve operacije iz relacijske algebre. Zahvaljujući takvim svojstvima, SQL je u izražajnom smislu ekvivalentan relacijskoj algebri.[42]

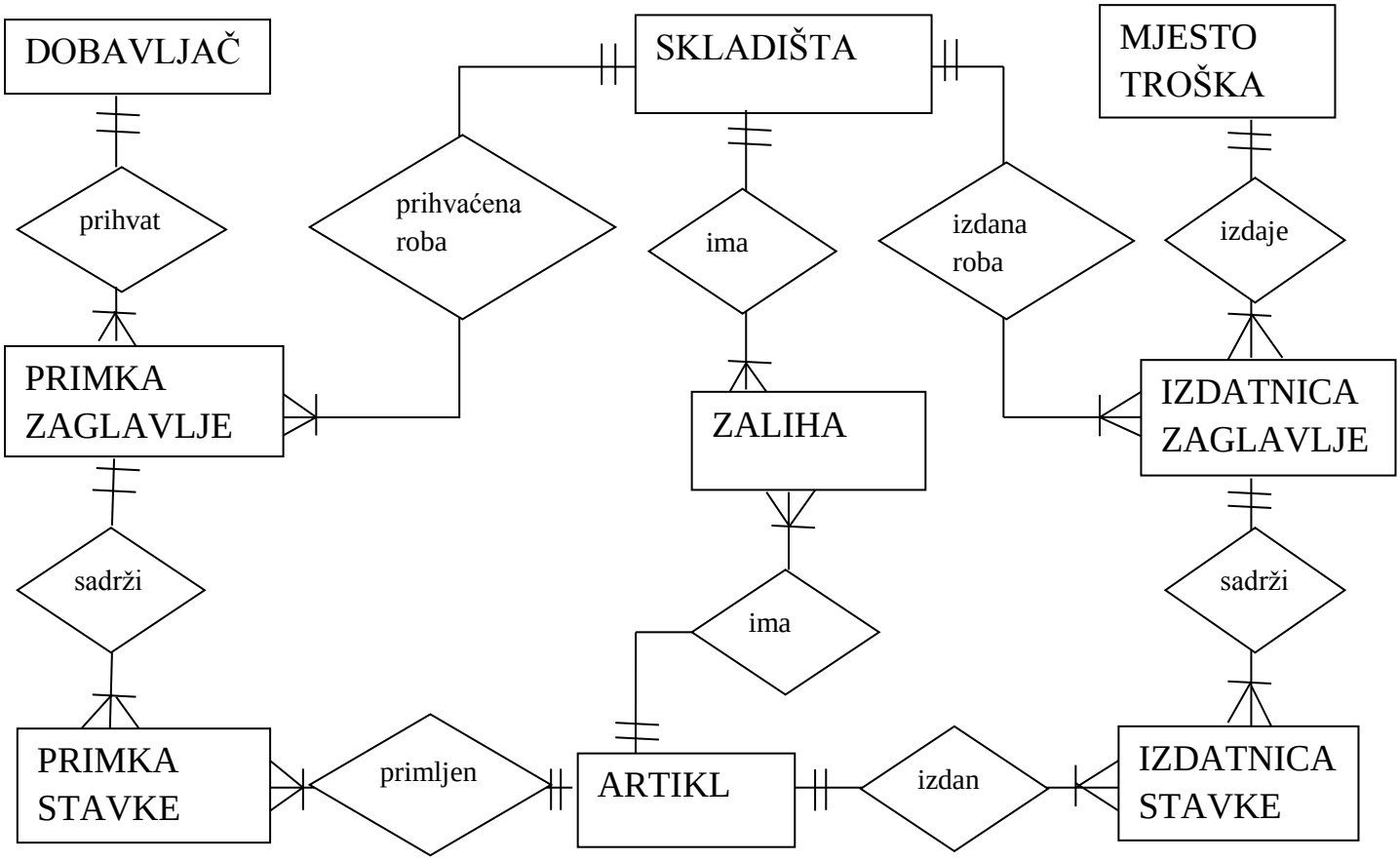
[41] Čerić Vlatko, Varga Mladen, op. cit., str. 387

[42] Manger Robert, op. cit., str. 84

4. PRIMJER MODELIRANJA PODATAKA POSLOVNOG PROCESA SKLADIŠTA

Prilikom modeliranja podataka poslovnog procesa prvo je potrebno izraditi konceptualni model (entiteti-veze) u ovom slučaju za skladište. Nakon izrade konceptualnog modela potrebno ga je implementirati u logički (relacijski) model gdje je izrađen pomoću alata MySQL Workbench. Na posljetku je potrebno postaviti nekoliko upita i analizirati dobivene rezultate.

4.1. Konceptualni model (entiteti-veze) skladišta



SLIKA 17: Konceptualni model (entiteti-veze) skladišta

(izvor: izradio student)

Slika 17 prikazuje konceptualni model (entiteti-veze) skladišta. Imena tipova entiteta odnosno, veza upisana su u odgovarajuće pravokutnike, odnosno rombove, te su povezani

spojnicama kako bi se znalo između kojih tipova entiteta postoji određena veza. Nakon izrade konceptualnog modela (entiteti-veze) potrebno je navesti atribute za sve entitete koji nisu vidljivi na shemi, također za svaki entitet označuje se primarni ključ koji je podvučenom punom linijom.

Tip entiteta DOBAVLJAČ ima atribute: ID, Naziv, Adresa, Mjesto.

Tip entiteta PRIMKA ZAGLAVLJE ima atribute: ID, Datum, Opis, Broj.

Tip entiteta PRIMKA STAVKE ima atribute: ID, Količina, Nabavna cijena.

Tip entiteta SKLADIŠTA ima atribute: ID, Naziv.

Tip entiteta ZALIHA ima atribute: ID, Količina.

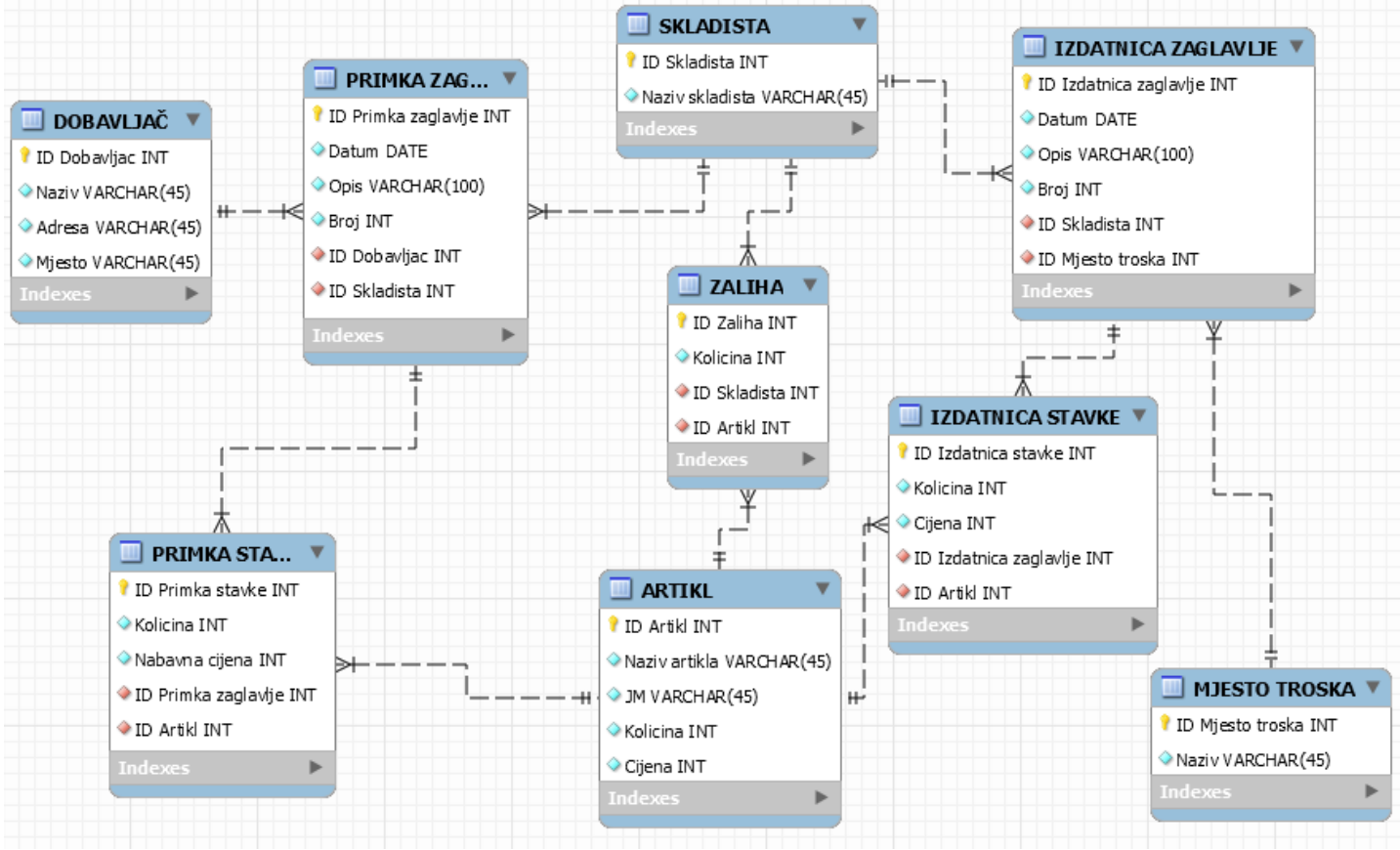
Tip entiteta ARTIKL ima atribute: ID, Naziv artikla, JM, Količina, Cijena.

Tip entiteta IZDATNICA ZAGLAVLJE ima atribute: ID, Datum, Opis, Broj.

Tip entiteta IZDATNICA STAVKE ima atribute: ID, Količina, Cijena.

Tip entiteta MJESTO TROŠKA ima atribute: ID, Naziv.

4.2. Relacijski model skladišta



SLIKA 18: Relacijski model skladišta

(izvor: izradio student)

NAPOMENA: Grafičko sučelje MySQL omogućuje crtanje isprekidanih linija koje nemaju posebno značenje.

Slika 18 prikazuje relacijski model skladišta koji je izrađen u alatu MySQL Workbench. Relacijski model sastoji se od 9 tablica koje su međusobno povezane. Svaku tablicu čini jedan entitet sa svojim atributima, te primarnim i stranim ključevima kako bi tablice bile međusobno povezane. Tablica Primka zaglavlje povezana je s tablicama Dobavljač i Skladišta preko stranih ključeva ID Dobavljač i ID Skladište, tablica Primka stavke povezana je s tablicama Primka zaglavlje i Artikl preko stranih ključeva ID Primka zaglavlje i ID Artikl, tablica Zaliha povezana je s tablicama Skladišta i Artikl preko stranih ključeva ID Skladišta i ID Artikl, tablica Izdatnica zaglavlje povezana je s tablicama Skladišta i Mjesto troška preko stranih ključeva ID Skladišta i ID Mjesto troška, te tablica Izdatnica stavke povezana je s tablicama Izdatnica zaglavlje i Artikl preko stranih ključeva ID Izdatnica zaglavlje i ID Artikl.

4.3. Rezultati upita

Na temelju kreiranog relacijskog modela postavljeni su sljedeći upiti:

SELECT ID, Naziv artikla, Cijena

FROM Artikl

WHERE Cijena < 10

ID	Naziv artikla	Cijena
666	Fanta	9
777	Čips	6

SLIKA 19. Rezultat upita

(izvor: izradio student)

Slika 19 prikazuje rezultat upita koji ispisuje sve artikle koji imaju cijenu manju od 10 kuna.

SELECT SKLADIŠTA. ID Skladišta, Naziv skladišta, ZALIHA. ID Zaliha, Količina, ARTIKL.ID Artikl, Naziv artikla

FROM SKLADIŠTA, ZALIHA, ARTIKL

WHERE ZALIHA.ID Skladišta=SKLADIŠTA.ID Skladišta AND ZALIHA.ID Artikl=ARTIKL.ID Artikl

ORDER BY ZALIHA.Količina DESC

ID Skladista	Naziv Skladista	ID Zaliha	Količina	ID Artikl	Naziv artikla
6	Skladiste pića	66	40	666	Fanta
5	Skladiste hrane 1	55	30	555	Pašta
7	Skladiste hrane 2	77	20	777	Čips

SLIKA 20. Rezultat upita

(izvor: izradio student)

Slika 20 prikazuje rezultat upita koji ispisuje nazive artikala i količine zaliha po skladištima, gdje se količina grupira od najveće prema najmanjoj.

SELECT IZDATNICA ZAGLAVLJE.ID Izdatnica zaglavlje, Datum, Opis, IZDATNICA STAVKE.ID Izdatnica stavke, Količina, Cijena, ARTIKL.ID Artikl, Naziv, MJESTO TROŠKA.ID Mjesto troška, Naziv

FROM IZDATNICA STAVKE, IZDATNICA ZAGLAVLJE, ARTIKL, MJESTO TROŠKA

WHERE IZDATNICA STAVKE.ID Izdatnica zaglavlje = IZDATNICA ZAGLAVLJE.ID Izdatnica zaglavlje AND IZDATNICA ZAGLAVLJE.ID Mjesto troška = MJESTO TROŠKA.ID Mjesto troška AND ARTIKL.ID Artikl = IZDATNICA STAVKE.ID Artikl

ORDER BY MJESTO TROŠKA.Naziv ASC

ID Izdatnica zaglavlje	Datum	Opis	ID Izdatnica Stavke	Količina	Cijena	ID Artikl	Naziv	ID Mjesto troška	Naziv
9	20.05.2015.	Isp.	99	30	9	666	Fanta	999	Kaufland
10	20.05.2015.	Isp.	100	30	6	777	Čips	1000	Konzum
8	20.05.2015.	Isp.	88	20	7	555	Pašta	888	Plodine

SLIKA 21. Rezultat upita

(izvor: izradio student)

Slika 21 prikazuje rezultat upita koji prikazuje popis artikala iz izdatnica po mjestu troška.

5.ZAKLJUČAK

Važnost informacija i informacijskog sustava neophodan je za uspjeh organizacije u poslovnom svijetu. Informacijski sustav osigurava sve potrebne informacije za obavljanje poslova i donošenja odluka. U prisutnosti sve veće globalizacije od iznimne važnosti su računalni programi bez kojih je nezamislivo raditi u organizacijama, koji omogućavaju zaposlenicima lakši pristup i lakše snalaženje među velikom količinom podataka. Tu se dolazi do važnosti baza podataka koje omogućavaju brže pretraživanje i organiziranje, odnosno uporabu podataka neophodnih za uspješno poslovanje. Ideja baze podataka je da svi podaci budu pohranjeni na jednom mjestu kako bi bili dostupni svim korisnicima i aplikacijskim programima u bilo koje vrijeme, te se stoga može zaključiti da je baza podataka zbirka zapisa pohranjenih na računalu. Na spomen baze podataka uglavnom se pomisli na relacijske baze podataka jer su one danas najzastupljenije.

Kako bi se izradila baza podataka potrebno je u konceptualnom modelu prikupiti i analizirati informacijske zahtjeve koji se obavljaju intervjuiranjem korisnika, istraživanjem dokumentacije i sl. Važna je komunikacija između projektanta i korisnika. Slijedom intervjuiranja i prikupljanja informacija izrađuje se model entiteti-veze koji mora biti razumljiv svim korisnicima. Nakon izrade konceptualnog modela (entiteti-veze) on se implementira u logički model. Ovaj model ne rezultira razradom konačne fizičke strukture podataka pa se naziva logičkim. On se opisuje u obliku sheme baze podataka. Za opis logičkog modela u ovom radu korišten je relacijski model. Modeliranje i dizajn podataka nikako ne završava ovdje. U ovom radu nije detaljno obrađeno fizičko modeliranje kao ni sigurnost i održavanje. Svaka od ovih tema zahtijeva da se posebno prouči i obradi kako bi se dobila zaokružena cjelina glede baza podataka.

6.LITERATURA

a) KNJIGE

1. Panian Željko, Poslovna informatika za ekonomiste, Potecon, Zagreb, 2005.

2. Manger Robert, Baze podataka, Element, Zagreb, 2012.

3. Čerić Vlatko,Varga Mladen, Informacijska tehnologija u poslovanju, Element, Zagreb, 2004.

4. Varga Mladen, Baze podataka, konceptualno, logičko i fizičko modeliranje, DRIP, Zagreb 1994.

b) INTERNET

1. http://e-ucenje.oet.unipu.hr/file.php/30/BP2012_4.pdf (10.2.2015.)

2. https://www.fkit.unizg.hr/_download/repository/Pred_BAZE_PR-2012-13_MV_konacno.pdf (10.2.2015.)

7. POPIS SLIKA

SLIKA 1: Koraci i poslovi pri oblikovanju baze podataka.....2

SLIKA 2: Prikaz entiteta.....4

SLIKA 3: Prikaz ključnih i neključnih atributa.....5

SLIKA 4: Chenov dijagram podataka za skladište.....8

SLIKA 5: Martinov dijagram podataka za skladište.....9

SLIKA 6: Barkerov dijagram podataka za skladište.....10

SLIKA 7: Prikaz relacijskog modela podataka..... 16

SLIKA 8: Prikaz relacije.....18

SLIKA 9: Operacija unije relacija artikl 1 i artikl 2.....22

SLIKA 10: Operacija presjeka relaciija artikl 1 i artikl 2.....23

SLIKA 11: Operacija razlike artikl 1 i artikl 2.....23

SLIKA 12: Operacija Kartezijevog produkta relacija dobavljač i artikl.....24

SLIKA 13: Operacija selekcije relacije dobavljač.....25

SLIKA 14: Operacija projekcije relacije dobavljač.....26

SLIKA 15: Operacija theta-spajanja relacija primka stavke i izdatnica stavke.....27

SLIKA 16: Operacija dijeljenja relacija dobavljač i uvjet.....28

SLIKA 17: Konceptualni model (entiteti-veze) skladišta.....34

SLIKA 18: Relacijski model skladišta…….....35

SLIKA 19: Rezultat upita.....36

SLIKA 20: Rezultat upita.....37

SLIKA 21: Rezultat upita.....37