

# Modeliranje entiteta i veza (entity - relationship modeling)

---

Vareško, Nensi

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://um.nsk.hr/um:nbn:hr:137:517920>

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

Download date / Datum preuzimanja: **2025-02-27**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli

Fakultet ekonomije i turizma

"Dr. Mijo Mirković"

**NENSI VAREŠKO**

**MODELIRANJE ENTITETA I VEZA (ENTITY - RELATIONSHIPS  
MODELING)**

Završni rad

Pula, 2016.

Sveučilište Jurja Dobrile u Puli

Fakultet ekonomije i turizma

"Dr. Mijo Mirković"

**NENSI VAREŠKO**

**MODELIRANJE ENTITETA I VEZA (ENTITY\_RELATIONSHIPS  
MODELING)**

Završni rad

JMBAG:0303016133, izvanredni student

Studijski smjer: Poslovna informatika

Predmet: Baze podataka

Mentor: Prof. dr. sc. Vanja Bevanda

Pula, rujan 2016.

## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Nensi Vareško, kandidat za prvostupnika Poslovne informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega 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, \_\_. \_\_. 2016.

\_\_\_\_\_

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
1. Informacijski sustav, razvoj i modeli.....	2
1.1. Razvoj informacijskog sustava.....	4
1.2. Modeli informacijskog sustava.....	9
2. Modeliranje podataka.....	12
2.1. Relacijsko (logičko) modeliranje.....	14
3. Baze podataka.....	18
4. Model entiteta i veza.....	23
4.1. Entitet.....	23
4.2. Veza.....	24
4.3. Atribut.....	24
4.4. Ključ.....	25
5. Koraci u otkrivanju entiteta i veza.....	27
6. Grafički prikaz ER modela.....	28
6.1. Batini, Ceri & Navathe.....	29
6.2. Chen.....	31
6.3. Elmasri & Navathe.....	33
6.4. Korth & Silberschatz.....	35
6.5. Martin.....	37
6.6. McFadden & Hoffer.....	39
6.7. Oracle's case method (Barker).....	41
7. Prošireni model entiteta i veza (The enhanced entity - relationship model (EER)).....	43
Zaključak.....	48
LITERATURA.....	49
POPIS SLIKA.....	50
Sažetak.....	51
Summary.....	52



## Uvod

Cilj je usporedba ekspresivnosti različitih grafičkih prikaza dijagrama entiteta - veze.

Modeliranje entiteta i veza je tehnika modeliranja podataka koja pomaže projektantima da naprave konceptualne modele. Konceptualno modeliranje je apstraktan prikaz poslovnih potreba koji se koristi za stvaranje baze podataka. Tehniku modeliranja entiteta i veza je razvio profesor Peter Chen. Dijagrami entiteta i veza zapravo pomažu u komunikaciji između projektanta i korisnika koja je često subjekt nesporazuma i to ovaj model uklanja. Hipoteza - model entiteta i veza moguće je opisati nizom tehnika koje se bitno ne razlikuju. U radu će se koristiti znanstvene metode apstrakcije, specijalizacije, komparativnu metodu, metodu modeliranja te će se konceptualno modelirati korisnički zahtjevi primjenom različitih tehnika.

Nakon uvoda, u prvom dijelu rada je opisan informacijski sustav, njegov razvoj te modeli. Drugo poglavlje govori o modeliranju podataka gdje opisujemo različite modele podataka, glavne faze oblikovanja podataka te relacijsko modeliranje. Treće poglavlje govori o bazama podataka gdje opisujemo arhitekturu baze podataka te njene prednosti. Četvrto poglavlje govori o modelu entiteta i veza gdje opisujemo entitet, vezu, atribut i ključ. Peto poglavlje govori o koracima u otkrivanju entiteta, veza i atributa. Šesto poglavlje prikazuje razne grafičke prikaze ER modela na primjeru skladišne primke. Sedmo poglavlje govori o proširenom modelu entiteta i veza gdje opisujemo podrazred i nadrazred, specijalizaciju i generalizaciju, njihove hijerarhije te nasljeđivanje i dane pojmove prikazane na EER dijagramu baze podataka sveučilišta.



## 1. Informacijski sustav, razvoj i modeli

„Bez podataka bilo bi nemoguće poslovati. U poslovanju se evidentiraju podaci o nizu elemenata poslovanja, kao što su nazivi i adrese poslovnih partnera ili klijenata, stanja gotovih proizvoda i sirovina u skladištima, količina naručene i prodane robe, izdani i primljeni računi itd. Podatke zapisujemo na tradicionalne medije kao što su papir ili film, a danas sve češće na elektroničke medije na kojima se ostvaruju široke mogućnosti suvremene elektroničke, odnosno informatičke obrade podataka. Ljudi imaju sposobnost stvaranja i razlikovanja pojmova (konceptata), spoznajnih jedinica kojima se služe u spoznaji svijeta. Na određeni način to su „jedinice znanja“. U informatičkoj teoriji i praksi, koja se bavi izgradnjom „sustava podataka“, često se služimo terminom objekt. Objekt je bilo što na što se može primijeniti pojam, odnosno koncept. Možemo reći da je sve o čemu pojmimo, na što se referiramo ili što opisujemo jest objekt tako dugo dok imamo pojam uz koji ga vezujemo. Primjerice moramo imati pojam posao da bismo mogli definirati i zapisati različite vrste (instance ili pojave) poslova. Da bismo komunicirali i zapisali pojmove služimo se podacima. Podatak je skup prepoznatljivih znakova, odnosno simbola zapisanih na nekom mediju, npr. papiru, filmu, magnetskom ili kojem drugom mediju. Pomoću podataka zapisujemo činjenice, pritom ne razmatramo ni njihovu interpretaciju ni njihov kontekst. Naš stvarni (poslovni) svijet opisujemo odabranim pojmovima, a zapisujemo ih podacima. U informatici se najjednostavniji elementarni podatak vezuje uz prikaz (zapis) jednog obilježja određenog objekta. Npr. podatak Ivan prikazuje obilježje imena jedne osobe. Informacija ili obavijest nastaje tumačenjem, odnosno interpretacijom podataka i stavljanjem u kontekst značenja. Dakle, protumačeni podatak donosi informaciju. Protumačeni elementarni podatak daje elementarnu informaciju. Elementarna informacija odnosi se na n - torqu <objekt, obilježje objekta, vrijeme, vrijednost obilježja>. Primjerice vrijednost 7,45 može biti protumačena ovako: objekt je tečajna lista, obilježje objekta je odnos euro/kuna, vrijednost obilježja je 7,45, a vrijeme je 3.1.2010. Elementarna informacija „3.1.2010. odnos euro/kuna jest 7,45“ ujedno predstavlja i elementarnu poruku. Informacija, odnosno obavijest jest

dakle činjenica s određenim značenjem. Ona donosi novost, obavještava o nečemu, otklanja neizvjesnost i općenito služi kao podloga za odlučivanje.“<sup>1</sup>

Podaci i informacije:

- Podaci predstavljaju sastavne dijelove informacije
- Informacija je izvedena obradom podataka
- Informacija služi za otkrivanje značenja podataka
- Točna, relevantna i pravodobna informacija je ključ dobrog donošenja odluka
- Dobro donošenje odluka je ključ organizacijskog preživljavanja u globalnom okruženju<sup>2</sup>

Informacijski sustav

„Informacijski sustav je sustav koji prikuplja, pohranjuje, čuva, obrađuje i isporučuje informacije važne za organizaciju, tako da budu dostupne i upotrebljive svakome kome su potrebne. Informacijski sustav se može, ali i ne mora, koristiti informacijskom tehnologijom.“<sup>3</sup>

U današnjoj ekonomiji koja je suočena s globalizacijom i novim tehnologijama, informacijski sustav kao potpora poslovanju dobiva sve više na značaju.

Informacijski sustavi mogu biti automatizirani (koriste baze podataka<sup>4</sup>) ili neautomatizirani (npr. dokumenti u papirnatom obliku).<sup>5</sup> Dva su osnovna razloga za automatizaciju: kompleksnost sustava i količina podataka.<sup>6</sup> Prednosti korištenja baza podataka su: dostupnost,

---

1 Varga, M., (2014), *Upravljanje podacima*, 2.izdanje,Zagreb: Element d.o.o., pp. 1-3

2 Prema: Cornel, C., Morris, S., Rob, P., (2013) *Database system, design, implementation and management*, 10th ed., USA: Joe Sabatino, P.6

3 Bosilj Vukšić, V. et al., (2004), *Informacijska tehnologija u poslovanju*, Zagreb: Element

4 Baza podataka je kolekcija podataka.

5 Prema: de Haan, L. et al., (2009), *Beginning Oracle SQL*, [CD], US: Apress, p. 1

6 Ibid.

raspoloživost, sigurnost i upravljivost. Pristup podacima omogućen je u svakom trenutku i u slučaju nepovoljnog događaja, podaci se mogu vratiti, te su zaštićeni pomoću kontrole pristupa i ovlastima. Baze podataka olakšavaju upravljanje velikim količinama podataka.<sup>7</sup>

## 1.1. Razvoj informacijskog sustava

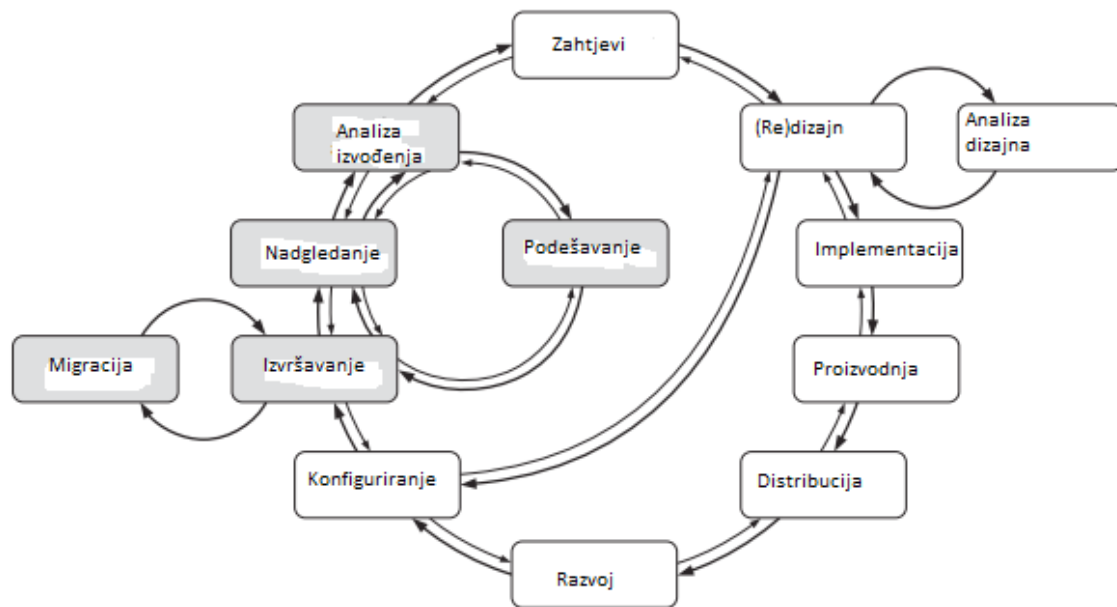
Postoje različiti načini za razvoj poslovnog informacijskog sustava. Model životnog ciklusa pokriva faze razvojnog procesa poslovnog informacijskog sustava. Poslovni informacijski sustavi poduzeća su kompleksni softverski sustavi koji su modificirani kako bi reflektirali organizacijske potrebe i promjene. Iz tog razloga, model životnog ciklusa uključuje faze koje se bave promjenom postojećih poslovnih informacijskih sustava. Informacijski sustavi uobičajeno imaju dva razvojna procesa. U prvom razvojnom procesu implementiran je općenit informacijski sustav; u drugom razvojnom procesu, taj sustav je prilagođen. Npr. za ERP<sup>8</sup> sustav, softverski prodavatelji, kao što je SAP, implementiraju nova izdanja svojih ERP sustava za druge organizacije. Implementacija njihovog ERP sustava je vođena razvojnim procesom softverskog prodavatelja. Nakon što organizacija kupi ERP sustav, taj ERP sustav prolazi kroz razvojni proces organizacije. U drugom razvojnom procesu ERP sustav treba biti instaliran, konfiguriran, prilagođen i predstavljen organizaciji. Može postojati mješavina ova dva razvojna procesa. Npr., informacijski sustav banke može biti sastavljen od odabranih komponenti ERP sustava i posebno razvijenih softverskih komponenti koje daju specifičnu funkcionalnost. U ovom slučaju, razvojni proces izrade informacijskog sustava za banku uključuje proces razvoja softvera sličan onome softverskih prodavatelja. Zbog njihove kompleksnosti, postojeći informacijski sustavi su obično redizajnirani i poboljšani prije nego što će biti zamijenjeni s novim sustavom. Kao posljedica, razvojni proces informacijskog sustava sadrži faze, kao što su održavanje i poboljšavanje. Npr., u informacijskom sustavu banke, sustav može biti rekonfiguriran i nadograđen na noviju verziju. Organizacije razvijaju i vode informacijske sustave što može uključivati softverske komponente nabavljene od drugih organizacija. Ljudi koji će koristiti informacijski sustav su korisnici ili

---

<sup>7</sup> Prema: de Haan, L. et al., (2009), *Beginning Oracle SQL*, [CD], US: Apress, pp. 1 - 2

<sup>8</sup> ERP (engl. Enterprise resource planning) je program poslovnog upravljanja koji se sastoji od raznih aplikacija koje organizacija koristi kako bi upravljala podacima i dobivala iz njih informacije.

sudionici. Ljudi koji dizajniraju informacijski sustav ili proizvode koji se koriste za sastavljanje informacijskih sustava su projektanti. Mnogi modeli životnog ciklusa su opisani u literaturi i korišteni u praksi. Neki teže ka procesu razvoja softvera, a drugi k razvoju informacijskog sustava u organizaciji.



Slika 1: Model životnog ciklusa informacijskog sustava poduzeća

Izvor:

[https://mitpress.mit.edu/sites/default/files/titles/content/9780262015387\\_sch\\_0001.pdf](https://mitpress.mit.edu/sites/default/files/titles/content/9780262015387_sch_0001.pdf)

Na slici 1 opisan je model životnog ciklusa. Svaki pravokutnik prikazuje fazu u životnom ciklusu, a lukovi prikazuju raspored faza. Glavni ciklus modelira razvojni proces novog informacijskog sustava. Uzima u obzir razvojni proces generičkog softvera, razvojni proces informacijskih sustava koji su prilagođeni od generičkog softvera i mješavina ta dva razvojna procesa. Dva manja ciklusa, koji sadrže osjenčane pravokutnike modeliraju razvojni proces postojećeg informacijskog sustava - održavanje i poboljšanje postojećih informacijskih sustava. Model životnog ciklusa prikazan na slici 1 je zasnovan na opažanju da su informacijski sustavi kompleksni, prilagođeni softverski sustavi čiji razvoj traje puno godina. Glavni ciklus na slici 1 specificira razvojni proces novog prilagođenog informacijskog sustava. Razlikujemo sljedećih jedanaest faza za prilagođeni informacijski sustav: faza zahtjeva, faza dizajna, faza analize dizajna, faza implementacije, faza proizvodnje, faza distribucije,

faza razvoja, faza konfiguracije, izvršna faza, faza nadgledanja i faza analize izvođenja. Nisu sve faze relevantne za sve informacijske sustave; npr., faze proizvodnje, distribucije i razvoja su relevantne samo u slučaju generičkih informacijskih sustava, kao što su ERP sustavi, Microsoft Office alati ili DBMS<sup>9</sup>. Faza zahtjeva je prva faza u životnom ciklusu na slici 1. Ona uključuje sakupljanje različitih zahtjeva za informacijski sustav i sastavljanje jedinstvene specifikacije zahtjeva. U mnogo slučajeva, postoji informacijski sustav koji ne zadovoljava sve zahtjeve. Onda je mudro temeljito analizirati što postojeći informacijski sustav čini za svoju okolinu. Rezultat ove analize će izvijestiti koje funkcionalnosti postojećeg informacijskog sustava treba zadržati u novom sustavu. Također, možemo dobiti vrijedan uvid analizirajući nedostatke postojećeg informacijskog sustava i razloge zbog kojih će novi informacijski sustav biti razvijen. Nakon ove analize, možemo formulirati zahtjeve novog informacijskog sustava. Namjena faze dizajna je razvoj dva modela koji su prikladni za komunikaciju s korisnicima i softverskim programerima informacijskog sustava. Prvo projektanti izvedu model funkcionalnog dizajna iz glavnog modela. Model funkcionalnog dizajna je izražen u uvjetima općenitog softverskog modeliranja. Nadalje, projektanti izvedu implementacijski model iz modela funkcionalnog dizajna uzimajući ciljni programski jezik ili implementacijski okvir u obzir. Model funkcionalnog dizajna obuhvaća funkcionalnost informacijskog sustava. Ovaj se model uobičajeno sastoji od više dijagrama koji predočavaju podatkovne modele i modele poslovnih procesa. Na taj način se odvaja od detalja implementacije. To je posebno važno za komunikaciju između korisnika i projektanta. Korisnici su laici i ne bi trebali biti upoznati sa svim detaljima informacijskog sustava; umjesto toga, krajnji korisnik mora razumjeti relevantne dijelove modela kako bi mogao znati da li je projektant pravilno formulirao zahtjeve. Projektanti mogu napraviti model funkcionalnog dizajna za informacijski sustav iz izvršne specifikacije pružajući formalni opis i prototip informacijskog sustava. Prototip je eksperimentalna prva verzija koja se koristi za testiranje i dizajn, te za dobivanje boljeg uvida u zahtjeve informacijskog sustava. To obično ne obuhvaća cijelu funkcionalnost informacijskog sustava. Izradom modela i eksperimentiranjem s prototipom, dvosmislenosti i skriveni zahtjevi mogu biti otkriveni. To osigurava da završni informacijski sustav zadovoljava zahtjeve svojih korisnika i izbjegava skupe i

---

<sup>9</sup> DBMS je sustav za upravljanje bazom podataka.

vremenski zahtjevne kasnije revizije. Drugi model, koji projektanti izvode tijekom faze dizajna, je implementacijski model. To je detaljan dizajn koji izvode programeri koji će implementirati informacijski sustav. Zato što je osnovno da implementacijski model odgovara modelu funkcionalnog dizajna, projektant mora provjeriti da li se ti modeli slažu. Uloga modela funkcionalnog dizajna i implementacijskog modela ne koristi se samo za diskusiju između projektanta i korisnika i između projektanta i programera. Analiza modela funkcionalnog dizajna i implementacijskog modela je tema treće faze životnog ciklusa, faze analize dizajna. Cilj ove faze je dobiti uvid u model i u informacijski sustav koji će biti implementiran. Ako je model apstraktan opis koji će se koristiti za analizu postojećeg sustava, model prvo mora biti provjeren. Provjera ispituje da li model ispravno reflektira informacijski sustav. Provjeren apstraktan opis modela i specifikacija modela mogu biti analizirani. Postoji više načina za analizu modela. Provjera je tehnika analiziranja koja pokazuje da li model zadovoljava specifikaciju. Specifikacija može biti drugi apstraktniji model ili skup svojstava koje model mora zadovoljiti. Druga korištena tehnika je simulacija. Ideja simulacije je napraviti izvršni model i na njemu provoditi određene pokuse (poznate kao scenarije). Simulacija je često primijenjena za analizu izvedbe. Četvrta faza u modelu životnog ciklusa je faza implementacije. U ovoj se fazi izrađuje informacijski sustav. Zbog toga što je informacijski sustav softverski sustav, izgradnja znači ili programiranje svih funkcionalnosti ili proširenje ili reimplementacija postojećih funkcionalnosti. Peta faza je faza proizvodnje u kojoj se softver informacijskog sustava priprema za distribuciju. Za razliku od klasičnog procesa proizvodnje, relativno je lako proizvesti softver, jer se to svodi na kopiranje i preuzimanje. U slučaju masovne proizvodnje, postoji i šesta faza, faza distribucije. Cilj ove faze je napraviti informacijski sustav dostupnim svojim budućim korisnicima. Marketing informacijskog sustava je također dio ove faze. U fazi razvoja, informacijski sustav je instaliran u svoju ciljanu okolinu i korisnici informacijskog sustava su podučeni kako ga koristiti. Brojne organizacije kupuju standardan softver. U tom slučaju, informacijski sustav mora biti konfiguriran i prilagođen organizaciji i njenim poslovnim procesima. Čak i kada organizacija razvija svoj vlastiti softver, često je potrebna konfiguracija. Za sektore kao što su financijsko računovodstvo, upravljanje zalihama ili planiranje proizvodnje, postoje standardizirani softverski paketi: ERP sustavi. Ti paketi imaju mnogo podesivih parametara. Na prilagodbu ERP sustava možemo gledati kao na programiranje u posebnom jeziku. Razlika od uobičajenog

programiranja je da sve funkcionalnosti ne moraju biti programirane. Prilagodba standardnih procesa specifičnoj organizaciji može zahtijevati znatan napor i ne smije biti podcijenjena. Nakon implementacije i konfiguracije, organizacije mogu napokon pokrenuti svoj informacijski sustav. U idealnom svijetu, izvršna faza bila bi završna faza razvojnog procesa. Zbog svoje kompleksnosti, mala je vjerojatnost da će informacijski sustav zadovoljavati sve zahtjeve i izvoditi se na način koji je očekivan na početku ove faze. Nadalje, okolina informacijskog sustava se mijenja tijekom vremena. Kako bi se pojednostavilo otkrivanje grešaka i kako bismo dobili uvid u to koje su zapravo funkcionalnosti korištene, informacijski sustavi bilježe veliki broj događaja. Te bilješke događaja daju detaljne informacije o aktivnostima koje su bile izvršene. Bilješke događaja imaju važnu ulogu u daljnjim fazama životnog modela. U desetoj fazi, fazi nadgledanja, organizacije izvlače informacije u stvarnom vremenu o performansama njihovog informacijskog sustava. Nadgledanjem dobivamo informacije o trenutnom stanju svakog poslovnog procesa i o performansama izvršenih aktivnosti. Faza nadgledanja je simulacija poslovnog procesa u praksi. Izvučene informacije mogu biti uspoređene s glavnim modelom. Za razliku od procesa pokazanog na slici 1, izvršna faza i faza nadgledanja se izvode paralelno. Nadgledanje se izvodi dok je informacijski sustav pokrenut, ali nije namjena promijeniti informacijski sustav ili redizajnirati poslovne procese. Nadgledanje daje relativno jednostavne dijagnostičke informacije. Za razliku od faze analize dizajna u kojoj se modeli informacijskog sustava analiziraju, faza analize izvođenja analizira da li informacijski sustav zadovoljava specifikacije. Bilješke događaja imaju važnu ulogu u ovoj fazi. Bilješke događaja mogu biti analizirane - npr. kako bi otkrili da li zahtjevi informacijskog sustava nisu ispoštovani. Tehnika rudarenja procesa omogućava da informacije budu izvučene iz bilješki događaja kako bi dali projektantima bolji uvid u informacijski sustav i poslovne procese.<sup>10</sup>

Uobičajena metoda procesa razvoja sustava poznatija pod nazivom „vodopad“ (engl. waterfall method) prikazuje faze razvojnog procesa i opise o tome što i kada treba učiniti.

---

10 Anonymous, (n.d.), Information Systems: Introduction and Concepts, [Internet], <raspoloživo na: [https://mitpress.mit.edu/sites/default/files/titles/content/9780262015387\\_sch\\_0001.pdf](https://mitpress.mit.edu/sites/default/files/titles/content/9780262015387_sch_0001.pdf)>, [pristupljeno 28.02.2016.]

Metoda „vodopada“<sup>11</sup>:

1. Analiza (opis potreba i određivanja granica sustava),
2. Logički dizajn (definiranje entiteta, atributa i stupnja integriteta),
3. Fizički dizajn (prijenos logičkog dizajna u stvarnu strukturu baze podataka),
4. Faza izgradnje (izgradnja aplikacija baze podataka).

## 1.2. Modeli informacijskog sustava

Modeli informacijskog sustava su: osobni informacijski sustav, poslovni informacijski sustavi (ERP, sustavi nabave, proizvodni sustavi, sustavi prodaje i marketinga, sustavi dostave, sustavi dizajna proizvoda, sustavi menadžmenta, skladišta podataka, sustavi poslovne inteligencije) i javni informacijski sustavi. Osobni informacijski sustavi upravljaju i skladište informacije privatne osobe. Primjeri su adresar, baza adresa ili kolekcija audio CD - a. Poslovni informacijski sustavi su napravljeni kako bi podržavali organizaciju. Javni informacijski sustavi upravljaju i skladište informacije kojima pristupa zajednica. ERP (engl. Enterprise Resource Planning, planiranje resursa poduzeća) sustav je informacijski sustav koji podržava glavni poslovni proces organizacije. Npr., upravljanje ljudskim resursima, prodaja, marketing, menadžment, financijsko računovodstvo, kontrola i logistika. U prošlosti je svaki poslovni proces bio ugrađen u zaseban informacijski sustav. Kako većina tih poslovnih procesa koristi povezane podatke, informacijski sustavi su morali skladištiti mnogo redundantnih podataka. Rastući broj i kompleksnost informacijskih sustava prisili su organizacije da ulože znatan trud u sinkroniziranje podataka svih informacijskih sustava. ERP sustav je rješenje za prevladavanje sinkronizacijskih napora integrirajući različite informacijske sustave. To je softverski sustav koji je izgrađen na raspodijeljenoj računalnoj platformi uključujući jedan ili više DBMS - a. Te računalne platforme služe kao infrastruktura na kojoj je svaki individualni poslovni proces implementiran. ERP sustav se izvodi na troslojnoj klijent/server arhitekturi koja se sastoji od sloja korisničkog sučelja, sloja aplikacijskog servera i sloja servera baze podataka. ERP

---

<sup>11</sup> Prema: de Haan, L. et al., (2009), *Beginning Oracle SQL*, [CD], US: Apress, p. 6



sustavi su u pravilu dizajnirani za multinacionalne kompanije. Sustav nabave je informacijski sustav koji pomaže organizaciji automatizirati proces nabave. Cilj sustava nabave je nabaviti sve što je potrebno kako bi se poslovni proces izvodio uz minimalne troškove. Od ključne važnosti je nabava prave količine materijala u pravo vrijeme i iz pravog izvora. Nabava je važan dio menadžmenta nabavnog lanca (engl. supply chain management). Menadžment nabavnog lanca ima za cilj koordiniranje i organizaciju dobavljača tako da se izbjegne neučinkovitost optimizirajući cijeli proces nabave. Proizvodni sustav podupire proizvodni proces u organizaciji. S povećanom automatizacijom proizvodnih procesa, proizvodni sustavi su postali sve važniji. Sustavi prodaje i marketinga trebaju obraditi klijentove narudžbe uzimajući u obzir pitanje dostupnosti. Organizacije poduzimaju promotivne aktivnosti i nude njihove proizvode po konkurentnim cijenama kako bi potaknuli prodaju. Jedan primjer promotivnih aktivnosti je kartica pogodnosti u supermarketima. Klijenti koji se prijave za karticu pogodnosti dobiju popust ili vaučer. Kartice pogodnosti su instrumenti organizacije kako bi dobili osobne podatke klijenata i podatke o njihovim kupovnim navikama. Te podatke sakuplja i obrađuje informacijski sustav. Informacije izvučene iz tih podataka mogu pomoći poboljšati marketing. Sustav dostave je informacijski sustav koji služi kao podrška dostave dobara klijentima. Zadaća ovih sustava je planiranje kada i kojim slijedom će korisnici dobiti svoje proizvode. Financijski sustavi održavaju tok novca u i između organizacija. Druga važna primjena financijskih sustava je burza.

Poslovni informacijski sustavi ne održavaju samo proizvodnju proizvoda, također održavaju i dizajn proizvoda. Primjeri su sustavi kompjuterski potpomognutog dizajna (engl. computer - aided design systems, CAD) i menadžment podataka proizvoda (engl. product data management systems, PDM). CAD sustavi održavaju grafičku prezentaciju i dizajn specifikacija proizvoda. PDM sustavi održavaju proces dizajna u širem smislu upravljajući dizajnom i dokumentacijom.

Sve više organizacija automatizira svoje poslovne procese. Organizacije moraju odrediti kojim redoslijedom aktivnosti poslovnog procesa morati biti izvršene i koja osoba mora izvršiti određenu aktivnost u određeno vrijeme. Tijek rada se odnosi na automatizaciju poslovnog procesa, u dijelu ili u cijelosti. Svaka aktivnost tijekom rada je implementirana u softver. Sustav upravljanja tijekom rada (engl. workflow management system) je informacijski sustav koji definira, upravlja i izvršava tijekom rada.

Skladište podataka je velika baza podataka koja skladišti bivše i trenutne informacije iz različitih izvora. Optimizirane su za brzo pretraživanje. Prvi proces izvlači podatke u pravilnim intervalima iz svojih informacijskih izvora, puni podatke u pomoćne tablice, čisti i obrađuje primljene podatke kako bi bili prikladni za shemu skladišta podataka. Obrada zahtjeva korisnika i analiza podataka je zadaća drugog procesa. Treći proces arhivira informacije koje više nisu potrebne. Upiti u skladištu podataka uobičajeno se odnose na poslovne događaje, kao što su prodajne transakcije ili posjete pri online trgovini koji su zapisani u tablicama povijesti događaja.

Sustav poslovne inteligencije pruža alate za analizu performansi poslovnog procesa. Ti alati izvlače informacije o poslovnim procesima iz podataka dostupnih organizaciji. Postoje različiti alati i tehnike, među kojima upravljanje poslovnim performansama, nadgledanje poslovnih aktivnosti, upiti i izvještavanje, rudarenje podacima i rudarenje procesima.<sup>12</sup>

---

<sup>12</sup> Ibid.

## 2. Modeliranje podataka

„Model podataka je skup pravila koji određuju kako sve može izgledati logička struktura baze podataka. Model čini osnovu za oblikovanje i implementaciju baze. Točnije rečeno, podaci u bazi moraju biti logički organizirani u skladu s onim modelom koji podržava odabrani DBMS.“<sup>13</sup>

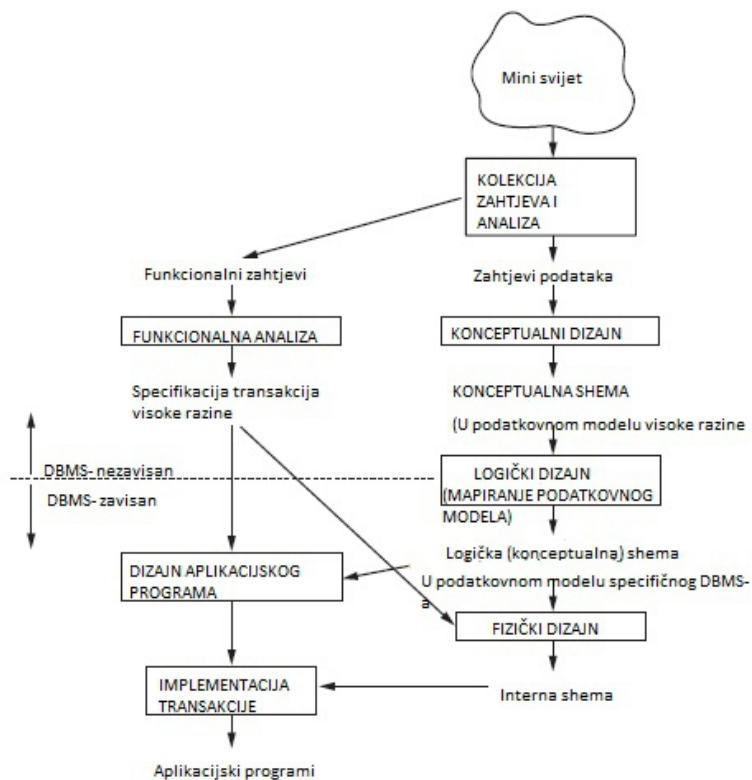
„Dosadašnji DBMS – i obično su podržavali neki od sljedećih modela:

- Relacijski model. Zasnovan je na matematičkom pojmu relacije. I podaci i veze među podacima prikazuju se tablicama koje se sastoje od redaka i stupaca.
- Mrežni model. Baza je predočena mrežom koja se sastoji od čvorova i usmjerenih lukova. Čvorovi predstavljaju tipove zapisa (slogova podataka), a lukovi definiraju veze među tipovima zapisa.
- Hijerarhijski model. Poseban slučaj mrežnog modela. Baza je predočena jednim stablom (hijerarhijom) ili skupom stabala. Svako stablo sastoji se od čvorova i veza „nadređeni - podređeni“ između čvorova. Čvorovi su tipovi zapisa, a odnos „nadređeni - podređeni“ izražava hijerarhijske veze među tipovima zapisa.
- Objektни model. Inspiriran je objektно orijentiranim programskim jezicima. Baza je predočena kao skup trajno pohranjenih objekata koji se sastoje od svojih internih „atributa“ (podataka) i „metoda“ (operacija) za rukovanje tim podacima. Svaki objekt pripada nekoj klasi. Između klasa se uspostavljaju veze nasljeđivanja, agregacije, te druge vrste veza.“<sup>14</sup>

---

13 Manager, R., (2012), *Baze podataka*, Zagreb: Element d.o.o., p. 2

14 Ibid., pp. 2-3



Slika 2: Glavne faze oblikovanja baze podataka

Izvor: Elmasri, R. & Navathe, S.B., (2011), *Fundamentals of database systems*, 6th ed., USA: Pearson education, p. 201

Slika 2 prikazuje dijagram s glavnim fazama oblikovanja baze podataka:

1. Prikupljanje i analiza zahtjeva (engl. Requirements collection and analysis)
2. Konceptualno modeliranje (engl. Conceptual design)
3. Logičko modeliranje (engl. Logical design)
4. Fizičko modeliranje (engl. Physical design)<sup>15</sup>

1. Prikupljanje i analiza zahtjeva - Sistemski analitičar, u suradnji s krajnjim korisnicima baze podataka, analizira poslovanje kako bi uvidio koji su podatkovni zahtjevi nove baze podataka. Uz analizu podatkovnih zahtjeva prikupljaju se i podaci o aktivnostima krajnjih korisnika baze kako bi se zadovoljila funkcionalnost aplikacije.<sup>16</sup>

<sup>15</sup> Elmasri, R. & Navathe, S.B., (2011), *Fundamentals of database systems*, 6th ed., USA: Pearson education, p. 201

<sup>16</sup> Ibid., pp. 201.-202.

2. Konceptualno modeliranje - Na osnovu podatkovnih zahtjeva izrađuje se konceptualna shema kojom se opisuju entiteti i veze među njima. Konceptualna shema je zbog svoje relativne jednostavnosti pogodna za komuniciranje s korisnicima i utvrđivanje mogućih propusta i konflikata. U ovoj fazi modeliranja razmatra same podatke, bez razmišljanja o načinu implementacije.<sup>17</sup> Konceptualni pogled je prikaz cijelog sadržaja informacija baze podataka u obliku koji je apstraktan u usporedbi s načinom na koji su podatci fizički pohranjeni. Različit je način na koji svaki određeni korisnik vidi podatke. Konceptualni pogled može se sastojati od zbirke zapisa o odjelu, zaposlenicima, dobavljačima itd. Konceptualni pogled definira konceptualna shema koja uključuje definicije o svakom od različitih konceptualnih tipova zapisa.<sup>18</sup>

3. Logičko modeliranje - Konceptualni model se prevodi u odgovarajući logički model, ovisno o DBMS - u. Kod relacijskih baza podataka, model se prevodi u relacije.<sup>19</sup>

4. Fizičko modeliranje - Tijekom ove faze unutarnja memorija se strukturira, organiziraju se datoteke, specificiraju se popisi, pristupni putevi i parametri fizičkog dizajna za baze podataka.<sup>20</sup>

Prilikom dizajniranja baze podataka, prije kodiranja same aplikacije, potrebno je prvo pravilno modelirati podatke, jer je naknadno ispravljanje takvih pogrešaka zahtjevno i predstavlja veliki trošak.<sup>21</sup>

## 2.1. Relacijsko (logičko) modeliranje

„Cilj logičkog oblikovanja je stvoriti relacijsku shemu baze, dakle shemu koja opisuje logičku strukturu baze u skladu s pravilima relacijskog modela podataka. Relacijski model bio je teorijski zasnovan još krajem 60 - ih godina 20. stoljeća, u radovima Edgara Codda. Zahvaljujući intenzivnom istraživanju te napretku samih računala, efikasnost relacijskih baza postupno se poboljšavala. Sredinom 80 - ih godina 20. stoljeća relacijski model postao je

---

<sup>17</sup> Ibid., pp. 201.-202.

<sup>18</sup> Prema: Date, C.J., (2003), *An introduction to database systems*, 8th ed., USA: Addison-Wesley, pp. 39-40

<sup>19</sup> Elmasri, R. & Navathe, S.B., (2011), *Fundamentals of database systems*, 6th ed., USA: Pearson education, p. 202

<sup>20</sup> Ibid., p. 202.

<sup>21</sup> Prema: de Haan, L. et al., (2009), *Beginning Oracle SQL*, [CD], US: Apress, p. 2

prevladavajući. Relacijski model zahtjeva da se baza podataka sastoji od skupa pravokutnih tablica - tzv. relacija. Svaka relacija ima svoje ime po kojem je razlikujemo od ostalih u istoj bazi. Jedan stupac relacije obično sadržava vrijednost jednog atributa (za entitet ili vezu) - zato stupac poistovjećujemo s atributom i obratno. Atribut ima svoje ime po kojem ga razlikujemo od ostalih u istoj relaciji. Dopušta se da dvije relacije imaju attribute s istim imenom, no tada se podrazumijeva da su to ustvari atributi s istim značenjem. Vrijednost jednog atributa su podaci iste vrste. Dakle, definiran je tip ili skup dopuštenih vrijednosti za atribut koji se također zove i domena atributa. Vrijednost atributa mora biti jednostruka i jednostavna (ne ponavljati se, ne da se rastaviti na dijelove). Pod nekim uvjetima toleriramo situaciju da vrijednost atributa nedostaje (nije upisana). Jedan redak relacije obično predstavlja jedan primjerak entiteta, ili bilježi vezu između dva ili više primjeraka. Redak nazivamo n - torka. U jednoj relaciji ne smiju postojati dvije jednake n - torke. Naime, relaciju tumačimo kao skup n - torki. Broj atributa zove se stupanj relacije, a broj n - torki je kardinalnost relacije. Kao primjer prikazana je relacija STUDENT, s atributima JMBAG, PREZIME, IME, GODINA STUDIJA. Relacija sadrži podatke o studentima koji su upisani na fakultetu.“<sup>22</sup>

„Ključ K relacije R je podskup skupa atributa od R sa sljedećim svojstvima:

1. Vrijednost atributa iz K jednoznačno određuje n - torku u R. Dakle, ne mogu postojati dvije n - torke s istim vrijednostima atributa iz K.
2. Ako izbacimo iz K bilo koji atribut, tada se narušava svojstvo 1.

Budući da su sve n - torke u R međusobno različite, K uvijek postoji. Naime, skup svih atributa zadovoljava svojstvo 1. Izbacivanjem suvišnih atributa doći ćemo do podskupa koji zadovoljava i svojstvo 2. Događa se da relacija ima više kandidata za ključ. Tada jedan od njih proglašavamo primarnim ključem. Atributi koji sastavljaju primarni ključ zovu se primarni atributi. Vrijednost primarnog atributa ne smije ni u jednoj n - torci ostati neupisana. Građu relacije kratko opisujemo tzv. shemom relacije. To je redak koji se sastoji od imena relacije te popisa imena atributa odvojenih zarezima i zatvorenih u zagrade. Primarni atributi su podvučeni. Npr. za relaciju o studentima, shema izgleda ovako:

STUDENT (JMBAG, PREZIME, IME, GODINA STUDIJA). Relacijska shema cijele baze zapisuje se tako da se nanižu sheme za sve relacije od kojih se ta baza sastoji. Dakle, shema baze ima onoliko redaka koliko u njoj ima relacija. Opisani prikaz relacijske sheme vrlo je koncizan i pregledan, no u njemu nedostaju informacije o tipovima atributa. Zato je potrebno

---

<sup>22</sup> Manger, R., (2012), *Baze podataka*, Zagreb: Element d.o.o., p.38

da se shema nadopuni rječnikom podataka, dakle popisom svih atributa, s pripadnim tipovima vrijednosti i neformalnim opisom. Tipovi ne moraju biti definirani onako kako će to zahtijevati fizička shema, nego onako kako to prirodno zahtijevaju sami podaci.“<sup>23</sup>

„Normalizacija je postupak kojim se iz danog modela baze podataka nastoji otkloniti potrebu za višestrukim zapisivanjem istih podataka. Definicije normalnih formi i postupaka normalizacije temelje se na ovisnostima (dependencies). Govorimo o trima vrstama ovisnosti: funkcijskoj (functional), višeznačnoj (multivalued) i spajanja (join). Neka X i Y budu podskupovi atributa relacije R. Na R vrijedi funkcijska ovisnost (FO).  $X \twoheadrightarrow Y$  ako za svaku instancu od X može postojati samo jedna instanca od Y. Kažemo da Y funkcijski ovisi od X, odnosno da X funkcijski determinira Y (X nazivamo i determinantom od Y). Npr., na shemi relacije KUPCI vrijedi FO  $IDKUP \twoheadrightarrow LIMIT$ . Naime, svaki kupac (identificiran šifrom  $IDKUP$ ) ima samo jedan limit. Iz istih razloga vrijedi i slijedeća FO:

$IDKUP \twoheadrightarrow STANJE, LIMIT$ . Primarni ključ ( $IDKUP$ ) funkcijski determinira svaki podskup atributa relacije kojoj pripada. Drugim riječima, kandidat ključa K relacije R može se definirati kao onaj (minimalni) podskup atributa od R koji funkcijski determinira sve attribute relacije R; dakle, kao onaj minimalni podskup za koji vrijedi:  $K \twoheadrightarrow R$ . Na temelju funkcijske ovisnosti definirane su prva, druga, treća i Boyce - Coddova normalna forma sheme relacije. Prema definiciji relacijske strukture, svaka relacija iz relacijskog modela jeste u prvoj normalnoj formi (1NF). 1NF zahtijeva da sve instance svih atributa budu atomarne vrijednosti. To drugim riječima znači da svaki atribut mora funkcijski ovisiti od primarnog ključa. Za definiranje ostalih triju NF valja najprije uvesti nekoliko (pomoćnih) pojmova. Atribut A sheme relacije R nazivamo ključnim (primarnim, prime) ako je sadržan u nekom od kandidata ključa relacije R. U suprotnome, atribut nazivamo neključnim (nonprime). Skup atributa S sheme relacije R nazivamo natključem (supreključem, superkey) od R ako S sadrži (obuhvaća) nekog od kandidata ključa iz R. Za drugu normalnu formu se obično kaže da je samo međukorak ka višim NF, da je zanimljiva samo u povijesnom smislu i nadasve da je nedovoljna za praksu. Drugim riječima, model baze podataka koji sadrži i takve sheme relacija koje su u 2NF (ali ne i višim NF) može (i treba) biti poboljšán tako da se te relacije nadomjeste novima koje su (barem) 3NF. Shema relacije R je u drugoj normalnoj formi (2NF) ako za svaku netrivialnu funkcijsku zavisnost  $X \twoheadrightarrow Y$  na R, vrijedi barem jedno:

- a) X nije pravi podskup od nijednog kandidata ključa ili
- b) Y je ključni atribut od R.

---

<sup>23</sup> Ibid. pp.38-39

Pogledajmo primjer: shema relacije STA1(IDNAR, IDPRO, DATUM, KOLIČ, PROCJE) nije u 2NF jer na toj shemi vrijedi FO IDNAR → DATUM. A atribut IDNAR (tj. X) jeste pravi podskup kandidata ključa (IDNAR, IDPRO), dok atribut DATUM (tj. Y) nije ključni atribut od R. Dakle shema relacije STA1 ne ispunjava nijednog od dvaju uvjeta iz definicije 2NF te nije u 2NF. Recimo to i na drugi (jednostavniji) način: Shema relacije STA1 nije u 2NF jer njen neključni atribut DATUM ovisi od dijela kandidata ključa (IDNAR, IDPRO). Shema relacije R je u trećoj normalnoj formi (3NF) ako za svaku netrivialnu funkcijsku ovisnost X → Y na R vrijedi barem jedno:

a) X je natključ od R ili

b) Y je ključni atribut od R.

NAR2 (IDNAR, DATUM, IZNOS, IDKUP, NAZIV, STANJE, LIMIT) jeste u 2NF jer čim nema sastavljen ključ, uvjet iz definicije 2NF je nužno ispunjen. Međutim, NAR2 nije u 3NF jer i sadrži i FO IDKUP → LIMIT, a niti je IDKUP natključ, niti je LIMIT ključni atribut od NAR2. Shema relacije NAR2 nije u 3NF jer njen neključni atribut LIMIT ovisi od drugog neključnog atributa IDKUP.

Shema relacije R je u Boyce - Coddovoj normalnoj formi (BCNF) ako za svaku netrivialnu funkcijsku ovisnost X → Y na R vrijedi: X je natključ od R. Uočimo da se BCNF razlikuje od 3NF samo po tome što izostavlja mogućnost (b) iz definicije 3NF i time zaoštava uvjet normalnosti forme. Npr. : shema relacije STAVKE (IDNAR, IDPRO, KOLIČ, PROCJE) jeste u BCNF jer su obje eksplicitno dane FO ujedno FO od natključa.<sup>24</sup>

---

24 Radovan, M, (1993), Baza podataka: Relacijski pristup i SQL, Zagreb: Informator, pp. 215-219



### 3. Baza podataka

„Baza podataka je skup međusobno povezanih podataka koji tvori zajedničku podatkovnu osnovu svih aplikacija nekog informacijskog sustava. Baza podataka je, s jedne strane temelj integralnosti (povezanosti, cjelovitosti) informacijskog sustava, a s druge strane omogućava neovisnost pojedinačnih aplikacija sustava od unutarnje organizacije podataka u bazi podataka.“<sup>25</sup>

„Baza podataka je skup međusobno povezanih podataka, pohranjenih u vanjskoj memoriji računala. Podaci su istovremeno dostupni raznim korisnicima i aplikacijskim programima.“<sup>26</sup>

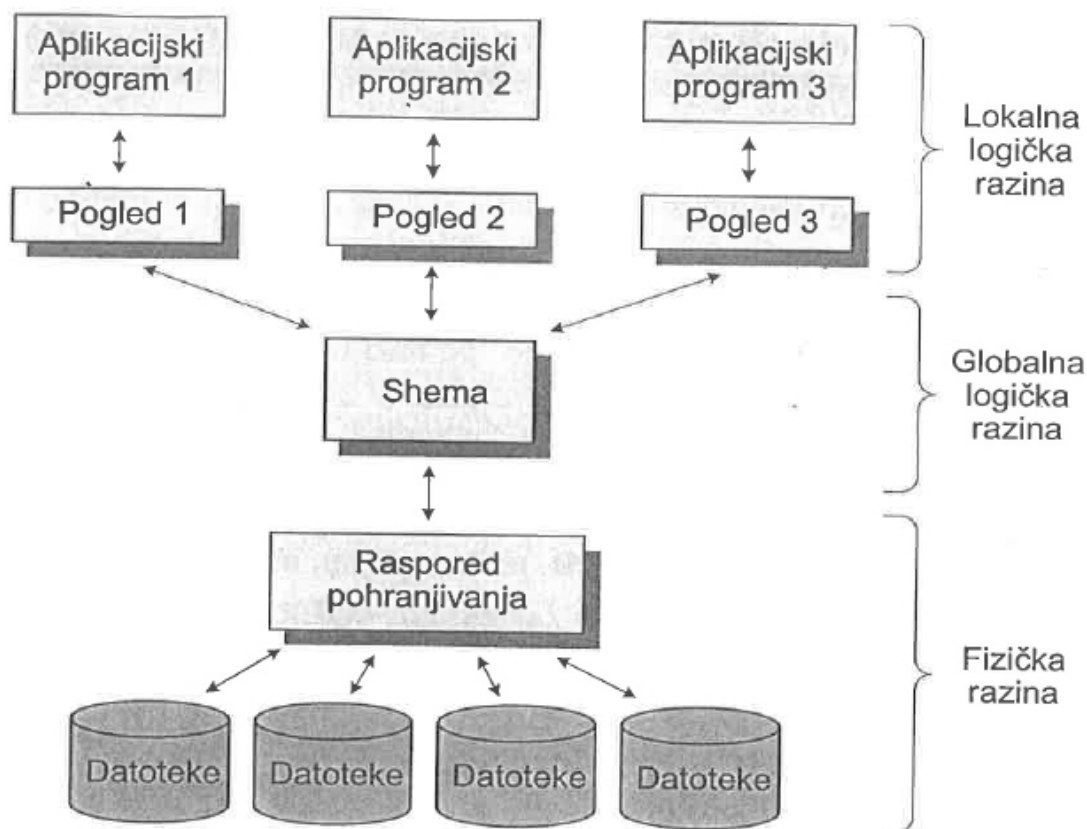
„Sustav za upravljanje bazom podataka (Data Base Management System - DBMS) je poslužitelj (server) baze podataka. On oblikuje fizički prikaz baze u skladu s traženom logičkom strukturom. Također, on obavlja u ime klijenata sve operacije s podacima. Dalje, on je u stanju podržati razne baze, od kojih svaka može imati svoju logičku strukturu, no u skladu s istim modelom. Isto tako, brine se za sigurnost podatka, te automatizira administrativne poslove s bazom.“<sup>27</sup>

---

25 Radovan, M, (1993), Baza podataka: Relacijski pristup i SQL, Zagreb: Informator, p. 1

26 Manger, R., (2012), *Baze podataka*, Zagreb: Element d.o.o., p. 1

27 Ibid., p. 1



Slika 3: Arhitektura baze podataka

Izvor: Manger, R., (2012), *Baze podataka*, Zagreb: Element d.o.o., p. 5

„Arhitektura baze podataka sastoji se od tri „sloja“ i sučelja među slojevima, kao što je prikazano na slici 3. Riječ je o tri razine apstrakcije:

- Fizička razina odnosi se na fizički prikaz i raspored podataka na jedinicama vanjske memorije. To je aspekt koji vide samo sistemski programeri (oni koji su razvili DBMS). Sama fizička razina može se dalje podijeliti na više podrazina apstrakcije, od sasvim konkretnih staza i cilindara na disku, do već donekle apstraktnih pojmova datoteke i zapisa (sloga) kakve susrećemo u klasičnim programskim jezicima. Raspored pohranjivanja opisuje kako se elementi logičke definicije baze preslikavaju na fizičke uređaje.

- Globalna logička razina odnosi se na logičku strukturu cijele baze. To je aspekt koji vidi projektant baze odnosno njezin administrator. Opis globalne logičke definicije naziva se shema (također engl. schema). Shema je tekst ili dijagram koji definira logičku strukturu baze, i u skladu je sa zadanim modelom. Dakle imenuju se i definiraju svi tipovi podataka i veze među tim tipovima, u skladu s pravilima rabljenog modela. Također, shema može uvesti i ograničenja kojima se čuva integritet podataka.
- Lokalna logička razina odnosi se na logičku predodžbu o dijelu baze koji rabi pojedina aplikacija. To je aspekt koji vidi korisnik ili aplikacijski programer. Opis jedne lokalne logičke definicije zove se pogled (view) ili podshema. To je tekst ili dijagram kojim se imenuju i definiraju svi lokalni tipovi podataka i veze među tim tipovima, opet u skladu s pravilima rabljenog modela. Također, pogled u svojoj konačnoj realizaciji zadaje i način na koji se iz globalnih podataka i viza izvode lokalni.<sup>28</sup>

### *Preslikavanja*

„Tri razine viđenja sadržaja baze, povezane su s dva preslikavanja (mappings) i to: konceptualno/unutarnje i vanjsko/konceptualno. Preslikavanja, s jedne strane, omogućavaju da se informacijski sadržaj s konceptualne razine (koji je oblikovan prema modelu podataka za dani organizacijski sustav) pohrani (na unutarnjoj razini) u obliku i na način koji je najpovoljniji s aspekta performansi sustava. S druge strane, preslikavanje omogućava da se, polazeći od konceptualne razine, oblikuju vanjski pogledi koji su primjereni potrebama i ovlastima pojedinih korisnika. Preslikavanje općenito omogućava da se za iste entitete koriste različita imena ali omogućava i formiranje novih entiteta. Npr. iz odabranih stupaca tablice s niže razine, može se formirati nova tablica na višoj razini. Analogno vrijedi i za stupce. Stupac s više razine može biti generiran funkcijski (tj. računanjem) iz više stupaca s niže razine. Tako definirane entitete nazivamo virtuelnima jer nemaju svog (vlastitog) zapisa na fizičkoj razini, već se izračunavaju (formiraju) iz drugog entiteta. Neovisnost programa od strukturnih promjena na razini baze temelji se na preslikavanjima. Pritom kažemo da se preslikavanjem vanjsko/konceptualno uspostavlja neovisnost programa

---

<sup>28</sup> Manger, R., (2012), *Baze podataka*, Zagreb: Element d.o.o., pp. 4-5

od logičke strukture baze (logical data independence), a da se preslikavanjem konceptualno/unutarnje uspostavlja neovisnost programa od fizičke strukture baze (physical data independence). Govoreći u praktičnim terminima, preslikavanje prvenstveno oslobađa zapise s više razine „tehničkih detalja“ svojstvenih nižoj razini definicije baze. Nadalje, preslikavanje omogućava upotrebu različitih imena za (fizički) iste entitete; konačno, definiranjem korisničkih pogleda (i virtuelnih tablica), sadržaj baze prilagođava se potrebama korisnika, a istodobno, ograničavanjem pristupa, sadržaj baze se i zaštićuje.<sup>29</sup>

Baza podataka otklanja slabosti koje su svojstvene aplikacijama sa zasebnim podatkovnim osnovama i otvara nove mogućnosti koje bi bez jedinstvene baze bile teško ostvarive.

*Kontrola redundance.* Datoteke koje pripadaju odvojenim aplikacijama često sadrže iste podatke: Dakle, sustav sadrži redundante podatke. A redundanca pored trošenja prostora – omogućava i pojavu proturječnih podataka u sustavu. Načelno govoreći, u bazi podataka svaki se podatak zapisuje samo na jedno mjesto.

*Integralnost sustava.* Baza podataka oblikuje se prema modelu podataka za određeni organizacijski (poslovni, tehnološki) sustav. Tako oblikovana baza podataka omogućava da informacijski sustav tvori cjelinu, odnosno da bude integralan. Naime, integralnost informacijskog sustava (kao sastavne cjeline) svodi se prvenstveno na mogućnost efikasnog protoka podataka među njegovim dijelovima (podsustavima).

*Korištenje zajedničkih podataka.* U načelu, sadržaj baze podataka dostupan je svakoj od aplikacija sustava. To ujedno znači da razvoj nove aplikacije ne iziskuje oblikovanje sasvim nove podatkovne osnove (za tu aplikaciju), već se i ona „veže“ na postojeću bazu podataka.

*Zaštita podataka.* Podaci su jedno od osnovnih sredstava na kojima se temelji djelovanje organizacijskih sustava. Neki su sustavi posebno osjetljivi na iskrivljenje, uništenje ili zloupotrebu podataka bilo zbog visokog tehnološkog rizika do kojeg mogu dovesti netočni podaci, bilo zbog tajnosti (ili vrijednosti) samih podataka. Stoga u svakom sustavu (mada ne u istoj mjeri) podatke treba štiti od iskrivljenja, uništenja i neovlaštene upotrebe. Na razini prava korištenja podataka, DBMS štiti sadržaj baze na taj način što pristup podacima dopušta samo onim korisnicima koji imaju primjerene ovlasti. Ovlasti se pritom odnose na točno određene podatke i na točno

---

29 Radovan, M, (1993), Baza podataka: Relacijski pristup i SQL, Zagreb: Informator, pp. 20-21

određene operacije nad tim podacima. Zaštita sadržaja baze od iskrivljenja temelji se na pravilima odnosno ograničenjima koja definira administrator baze podataka. DBMS ne dopušta izvođenje onih operacija koje nisu u skladu s danim pravilima, odnosno čijim bi izvršenjem bila prekršena dana ograničenja.

*Standardizacija podataka.* Baza podataka olakšava usvajanje određenih standarda kako u izboru tako i u oblikovanju podataka. Standardizirati se može, npr., način šifriranja pojedinih elemenata sustava (naprimjer: proizvoda, usluga i sličnog) ali i način oblikovanja (i obrade) određenih podataka.

*Optimizacija cjeline.* Informacijski sustav nalazi se u funkciji nekog tehnološkog, poslovnog ili organizacijskog sustava. U cilju optimalnog djelovanja cjeline, u takvim je sustavima potrebno usklađivati djelovanje (i tendencije) pojedinih podsustava, odnosno prilagođavati pojedine dijelove sustava potrebama (i mogućnostima) drugih dijelova. Postojanje jedinstvenog modela (i baze) podataka olakšava otklanjanje nesklada i dostizanje optimalnog djelovanja cjeline.<sup>30</sup>

---

30 Ibid. pp. 3-5

## 4. Model entiteta i veza

Chen je 1977. predstavio model koji pomoću dijagrama prikazuje stvarni svijet. E - R model koristi prirodniji pogled na svijet koji se sastoji od entiteta i veza.<sup>31</sup>

Model entiteta i veza dozvoljava da opišemo podatke o poduzeću iz stvarnog svijeta u oblicima objekata i njihovih veza i u širokoj je uporabi u razvoju početnog projekta baze podataka. Daje korisne koncepte koji nam omogućavaju da se krećemo od neformalnog opisa od toga što korisnici žele od svoje baze podataka do detaljnijeg opisa koji može biti implementiran u DBMS.<sup>32</sup>

### 4.1. Entitet

„Entitet je nešto što postoji i što se u stvarnom svijetu može identificirati.“<sup>33</sup>

„Entitet je stvaran ili apstraktan predmet ili događaj o kojem se u informacijskom sustavu prikupljaju podaci.“<sup>34</sup>

„Entitet je konkretna ili apstraktna posebnost, koncept ili objekt od interesa. On se po svojim obilježjima razlučuje od okolice.“<sup>35</sup>

Svijet je stvoren od entiteta. Entiteti mogu biti razvrstani u tipove entiteta, npr. zaposlenik je pojava općenitog tipa entiteta ZAPOSLENIK. Prednost takvog razvrstavanja je da će entiteti danog tipa imati zajedničke karakteristike (npr. plaća). Svaki entitet ima posebno svojstvo koje služi za identifikaciju tog entiteta, odnosno svaki entitet ima svoj identitet. Entiteti su povezani s drugim entitetima putem veza. Primjeri entiteta: DOBAVLJAČ, DIO, ZAPOSLENIK, ODJEL, OSOBA, KOMPOZICIJA, KONCERT, ORKESTAR, itd.<sup>36</sup>

---

31 Prema: Chen, P., The entity-relationship model—toward a unified view of data, ACM Transactions on Database Systems (TODS) - Special issue: papers from the international conference on very large data bases: September 22–24, 1975, Framingham, Volume 1 Issue 1, p. 1

32 Prema: Ramakrishnan, R. & Gehrke, J., (2003), *Database management systems*, 3rd ed., New York: McGraw-Hill, p.25

33 Varga, M., (1994), *Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka*, Zagreb: DRIP, p. 45

34 Ibid.

35 Ibid.

36 Date, C.J., (2003), *An introduction to database systems*, 8th ed., USA: Addison-Wesley, pp. 411-413

Postoje jaki entiteti i slabi entiteti. Slabi entiteti su entiteti kojima je postojanje zavisno o nekom drugom entitetu, u tom smislu da ne može postojati bez da postoji drugi entitet.<sup>37</sup>

## 4.2. Veza

„U stvarnom svijetu interesantan je međusoban odnos entiteta. Taj se odnos opisuje vezom. Veza predstavlja agregaciju dvaju ili više entiteta u novi entitet - vezu.

Broj entiteta koji sudjeluju u vezi jest stupanj veze. Veze stupnja 2, 3 i n zovu se binarnom, ternarnom i n - arnom. Prikazuju se odgovarajućom binarnom, ternarnom i n - arnom relacijom. Poseban slučaj binarne veze u kojoj na obje strane veze sudjeluju pojave istog tipa entiteta zove se unarnom (rekurzivnom ili involucijskom) vezom.<sup>38</sup>

Npr. postoji veza koja se zove DEPT\_EMP između odjela i zaposlenika, koja predstavlja činjenicu da je određeni odjel rukovođen od strane određenih zaposlenika. Za entitete dane u vezi kažemo da su sudionici u toj vezi. Neka R bude tip veze koji uključuje tip entiteta E kao sudionika. Ako svaka pojava od E sudjeluje u barem jednoj pojavi od R, onda je sudjelovanje od E u R potpuno, inače se kaže da je djelomično. Npr. ako svaki zaposlenik mora pripadati jednom odjelu onda je sudjelovanje zaposlenika u DEPT\_EMP potpuno; ako je moguće za dani odjel da nema zaposlenika, onda je sudjelovanje odjela u DEPT\_EMP djelomično. Veze mogu biti kardinalnosti jedan – naprama - jedan, jedan – naprama – više ili više – naprama -više.<sup>39</sup>

## 4.3. Atribut

„Svaki tip entiteta opisan je skupom atributa (obilježja ili svojstava).

Svaki atribut može poprimiti jednu vrijednost iz domene vrijednosti atributa.

Domena, dakle, definira moguće vrijednosti atributa. Formalno, atribut je funkcija koja skupu entiteta pridružuje skup vrijednosti odgovaraju

e domene. Veza može, ali ne mora, biti opisana atributima.<sup>40</sup>

---

37 Ibid, p. 414.

38 Varga, M., (1994), Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, Zagreb: DRIP, p. 47

39 Date, C.J., (2003), An introduction to database systems, 8th ed., USA: Addison-Wesley p. 416.

40 Varga, M., (1994), Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, Zagreb: DRIP, p.53

Npr. entitet ZAPOSLENIK može biti opisan zaposlenikovim Imenom, Godinama, Adresom, Plaćom i Poslom.<sup>41</sup>

Složeni atribut može biti podijeljen na manje poddijelove, što predstavlja jednostavnije attribute s nezavisnim značenjem. Npr. atribut Adresa kod entiteta ZAPOSLENIK može biti podijeljena na Adresa ulice, Grad, Državu i Poštanski broj s vrijednostima Zagrebačka 9, Pula, Hrvatska i 52100. Složeni atributi mogu tvoriti hijerarhiju; npr. Adresa ulice može dalje biti podijeljena na tri jednostavna atributa: Broj, Ulica i Broj\_stana. Vrijednost složenog atributa je ulančavanje vrijednosti od komponenti jednostavnih atributa. Složeni atributi su korisni za modeliranje situacija u kojima se korisnik ponekad odnosi na složeni atribut kao jedinicu ali u drugo vrijeme se odnosi specifično na neku njegovu komponentu. Jedna osoba može ne imati završen fakultet, druga osoba može imati završen jedan, a treća osoba može imati završena dva ili više fakulteta; nadalje različiti ljudi mogu imati različit broj vrijednosti za atribut Završeni\_fakultet. Takve attribute zovemo viševrijednosnim atributima. Viševrijednosni atribut može imati donju i gornju granicu kao ograničenje broju vrijednosti dopuštenih za svaki individualan entitet. Npr. atribut Boja entiteta AUTO može biti ograničen na između dvije ili tri vrijednosti, ako pretpostavimo da auto može imati najviše tri boje. U nekim slučajevima, dvije ili više vrijednosti atributa su povezane - npr. atributi Godine i Datum\_rođenja entiteta OSOBA. Za određeni entitet OSOBA vrijednost godina može biti određena iz današnjeg datuma i vrijednosti Datum\_rođenja te osobe. Atribut Godine zovemo izvedeni atribut.<sup>42</sup>

#### 4.4. Ključ

„Za identifikaciju pojave entiteta koristi se koncept ključa. Pojave entiteta međusobno se razlikuju po vrijednostima svojih atributa, pa je dovoljno pronaći takav skup atributa (jedan ili više) čije vrijednosti omogućuju jednoznačnu identifikaciju svake pojave entiteta u skupu entiteta – to je 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 (odnosno kandidatnim) ključevima. Tako npr. matični broj građanina te broj osobne iskaznice i mjesto izdavanja osobne iskaznice dva su moguća ključa. Jedan izabrani ključ postaje primarnim ključem. Superključem se naziva

<sup>41</sup> Elmasri, R. & Navathe, S.B., (2011), *Fundamentals of database systems*, 6th ed., USA: Pearson education, p.203.

<sup>42</sup> Ibid., p. 206.



bilo koji skup atributa koji sadrži ključ. On, dakle, ne mora biti minimalan. Identifikacijski slabi entiteti nemaju dovoljno atributa za jednoznačnu identifikaciju svake pojave entiteta, pa se ne može formirati primarni ključ. U skupu škola se npr. atributom Oznaka\_razreda (čije vrijednosti mogu biti npr 4a, 8b) ne mogu identificirati pojave identifikacijski slabog entiteta RAZRED. One se mogu identificirati ako se zna škola kojoj razred pripada. Stoga se slabi entitet RAZRED povezuje s jakim entitetom Škola vezom tipa 1,M:1,1 ili 0,M:1,1.

Vezom se svaka pojava slabog entiteta povezuje s jednom pojavom jakog entiteta. Iako identifikacijski slab entitet sam po sebi nema primarni ključ, potrebno je jednoznačno identificirati pojave slabih entiteta koji pripadaju pojavi jakog entiteta (npr. pojave RAZREDA koje pripadaju istoj ŠKOLI). Skup atributa slabog entiteta koji omogućuje ovu identifikaciju naziva se diskriminatorom (npr. diskriminator RAZREDA je Oznaka\_razreda). Primarni ključ identifikacijski slabog entiteta stoga se formira od primarnog ključa jakog entiteta i diskriminatora slabog entiteta (primarni ključ RAZREDA je kombinacija primarnog ključa ŠKOLE, to je Naziv\_Škole, i diskriminatora, to je Oznaka\_razreda).“ (Varga, 1994,p. 53)

## 5. Koraci u otkrivanju entiteta, veza i atributa

„Prvi je korak u oblikovanju konceptualne sheme otkrivanje samih elemenata od kojih se ta shema sastoji, a to su entiteti, veze i atributi. U pravilu, elementi sheme trebaju se prepoznati čitanjem specifikacije. Analiziraju se rečenice iz specifikacije i uočavaju imenice (subjekti, objekti) i glagoli (predikati).

- Imenice upućuju na entitete i attribute
- Glagoli upućuju na veze<sup>43</sup>

„U prepoznavanju elemenata sheme, projektant se često susreće s dilemom treba li neku važnu imenicu iz specifikacije shvatiti kao entitet ili kao atribut. Dilema se rješava odgovorom na sljedeća pitanja:

- Ima li pojam označen imenicom neka svoja dodatna svojstva koja treba pamtiti?

Ako da, onda je to entitet.

- Inače, jeli taj pojam svojstvo koje može poprimiti više vrijednosti kad njime opisujemo neki predmet, osobu ili pojavu? Ako da, onda je to opet entitet.
- Inače je taj pojam atribut.

Nakon što smo otkrili entitete, veze i attribute, potrebno je za svaki entitet utvrditi koji atributi ga opisuju. Ne treba zaboraviti mogućnost da neki atributi pripadaju vezi između entiteta, a ne pojedinim entitetima. Također za svaku vezu treba odrediti njezinu funkcionalnost, obaveznost članstva te kardinalnosti. Poželjno je da za svaki atribut imamo neku približnu

---

43 Manger, R., (2012), *Baze podataka*, Zagreb: Element d.o.o., p. 29

predodžbu o tipu vrijednosti koje on može poprimiti, makar se u ovoj fazi još ne zahtijeva da taj tip bude točno određen. Dalje, za svaki entitet treba odabrati primarni ključ.“ (Ibid.)

„Nakon što smo otkrili sve entitete, veze i atribute, idući korak u oblikovanju konceptualne sheme jest da se ti elementi povežu i prikažu u obliku dijagrama.“<sup>44</sup>

„Zadnji je korak u oblikovanju konceptualne sheme sastavljanje teksta koji prati dijagram.“<sup>45</sup>

## 6. Grafički prikazi ER modela

Ovdje ćemo razmotriti neke od prikupljenih ER modela na primjeru skladišne primke.

Objašnjenje dijagrama:

DOBAVLJAČ zaprima jednu ili više NARUDŽBENICA, NARUDŽBENICA je zaprimljena od jednog i samo jednog DOBAVLJAČA. NARUDŽBENICA sadrži jednu ili više STAVKI\_PRIMKE, STAVKA PRIMKE je sadržana u jednoj i samo jednoj NARUDŽBENICI. STAVKA\_PRIMKE sadrži jedan ili više ARTIKALA, ARTIKAL je sadržan u jednoj i samo jednoj NARUDŽBENICI. ARTIKAL može biti ROBA ili MATERIJAL. ARTIKAL sadrži jednu ili više AMBALAŽA\_ARTIKLA, AMBALAŽA\_ARTIKLA sadrži nijedan ili više ARTIKALA. AMBALAŽA\_ARTIKLA sadrži jednu ili više AMBALAŽA, AMBALAŽA sadrži jednu ili više AMBALAŽA\_ARTIKLA. ARTIKAL ima atribute Naziv, Jed. Mjere, Količina, Skladišni broj, Cijena i Iznos. ARTIKAL doprema jedan i samo jedan NAČIN\_DOPREME, NAČIN DOPREME doprema jedan ili više ARTIKLA. NAČIN\_DOPREME može biti vagon, automobil ili drugo. NAČIN\_DOPREME ima atribut Oznaka i broj vagona, automobila i drugo. ARTIKAL je jedna i samo jedna STAVKA\_PRIMKE, STAVKA\_PRIMKE je jedan ili više ARTIKALA. STAVKA\_PRIMKE sadrži jednu i samo jednu PRIMKU, PRIMKA sadrži jednu ili više STAVKI\_PRIMKE. PRIMKA ima atribute Datum i Broj. PRIMKU zaprima jedno i samo jedno SKLADIŠTE, SKLADIŠTE zaprima jednu ili više PRIMKI. PRIMKU knjiži jedna i samo jedna OSOBA, OSOBA knjiži jednu ili više PRIMKI. OSOBA

---

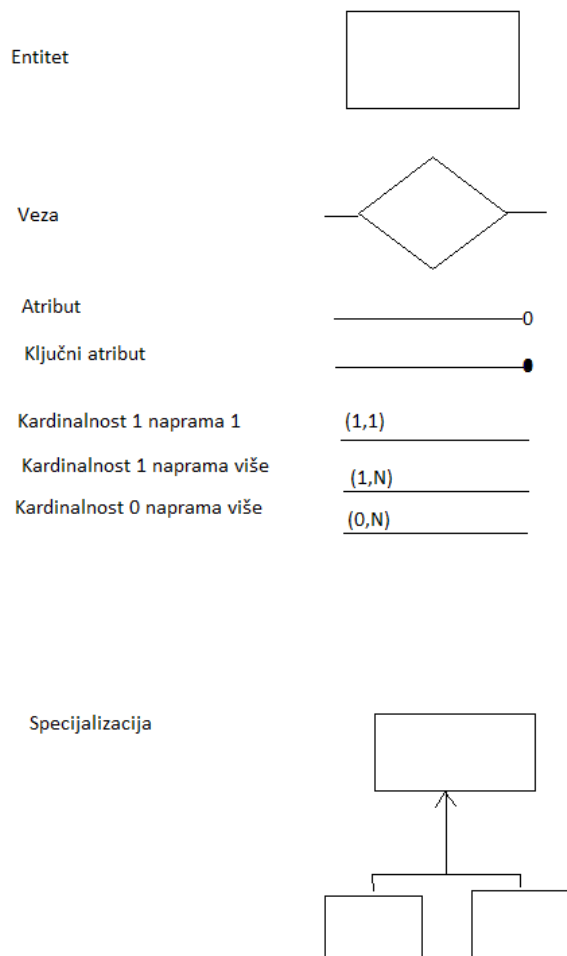
44 Ibid., p. 28

45 Ibid., p. 29

ima jednu ili više VRSTA\_POSLA, VRSTA\_POSLA ima jednu i samo jednu osobu.  
VRSTA\_POSLA ima attribute Šifra posla i Opis posla.

### **6.1. Batini, Ceri & Navathe**

Osnovni elementi grafičkog prikaza Batini i ostali su:



Slika 4: Batini, Ceri & Navatheov prikaz

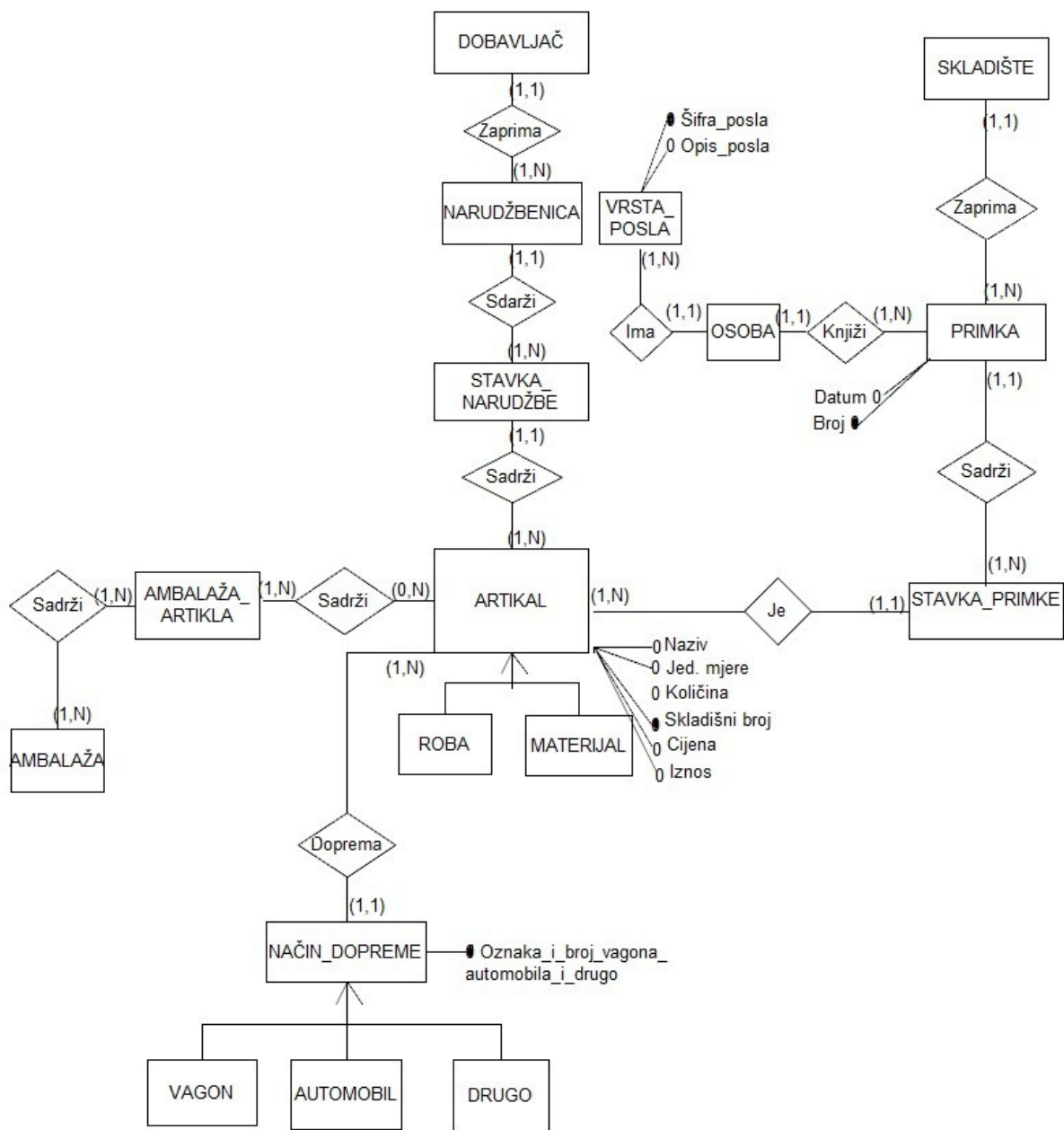
Izvor: Izradila studentica<sup>46</sup>

Slika 4 Prikazuje osnovne komponente modela Batini, Ceri & Navathe

Dozvoljava n - arne veze. Na ERD razini ne dozvoljava vanjski ključ.<sup>47</sup>

46 Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, p. 15

47 Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, pp. 31-32



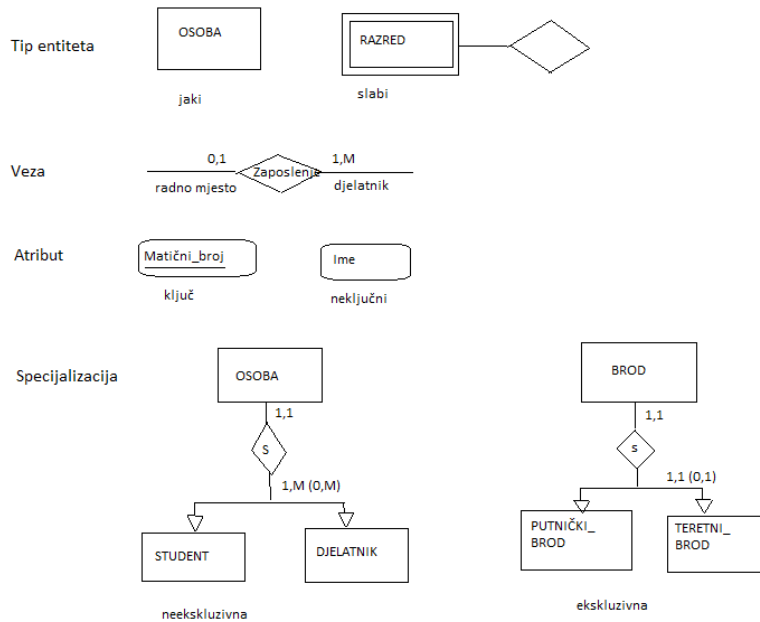
Slika 5: ER dijagram Batini, Ceri & Navathe modelom

Izvor: Izradila studentica

Slika 5 prikazuje ER dijagram na slučaju skladišne primke.

## 6.2. Chen

Osnovni elementi Chenovog grafičkog prikaza su:



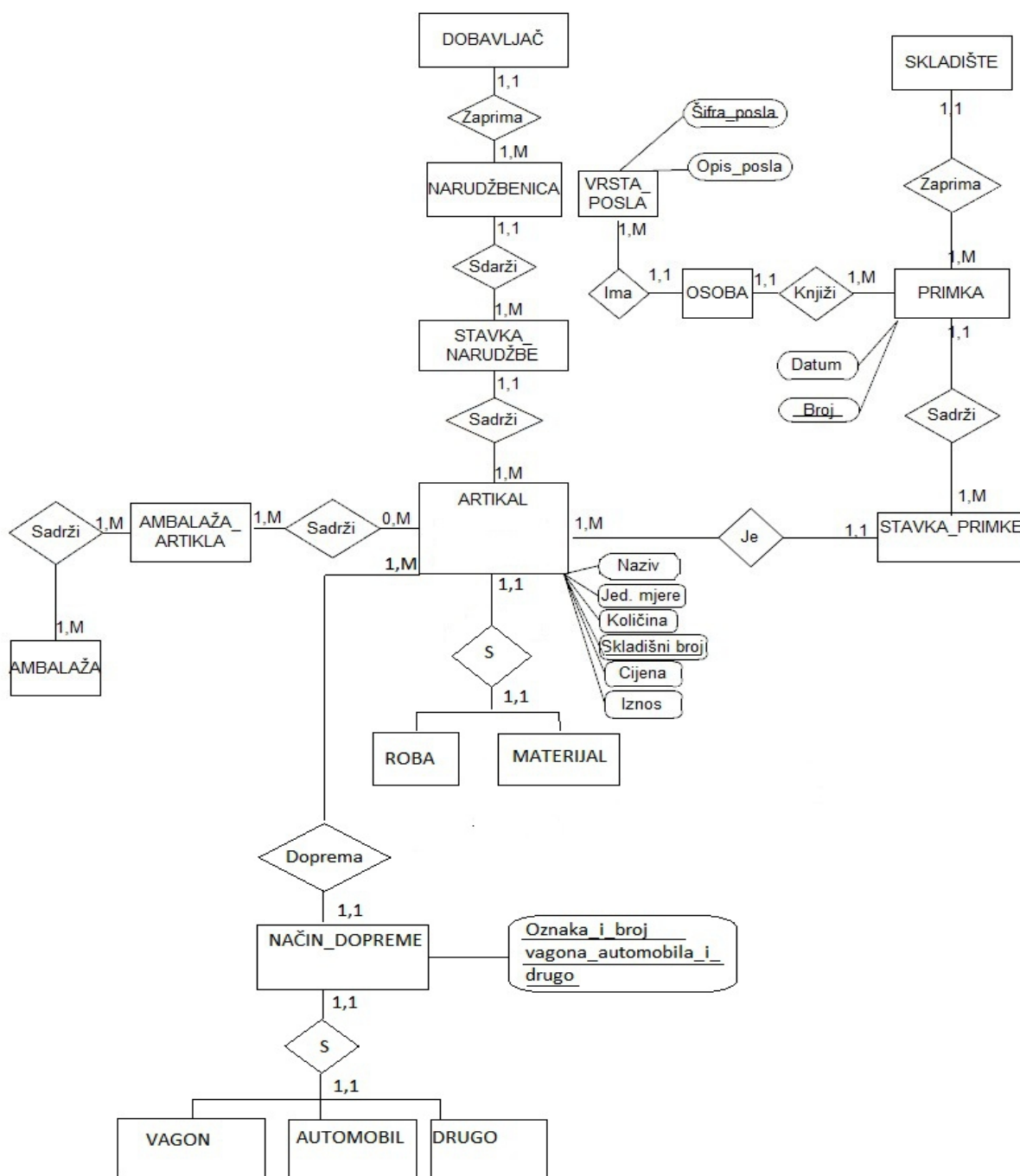
Slika 6: Chenov prikaz

Izvor: Varga, M., (1994), Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, Zagreb: DRIP, p.56

Slika 6 prikazuje osnovne komponente Chenova modela

Dozvoljava n - arne veze. Na ERD razini ne dozvoljava vanjski ključ.<sup>48</sup>

48 Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, pp. 31-32



Slika 7: ER dijagram Chenov model

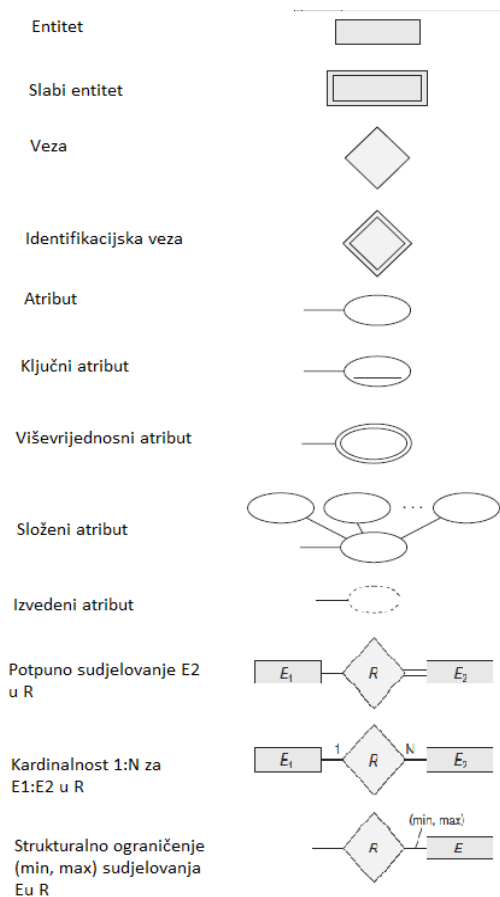
Izvor: Izradila studentica

Slika 7 prikazuje ER dijagram na slučaju skladišne primke.



### 6.3. Elmasri & Navathe

Osnovni elementi grafičkog prikaza Elmasri & Navathe su:



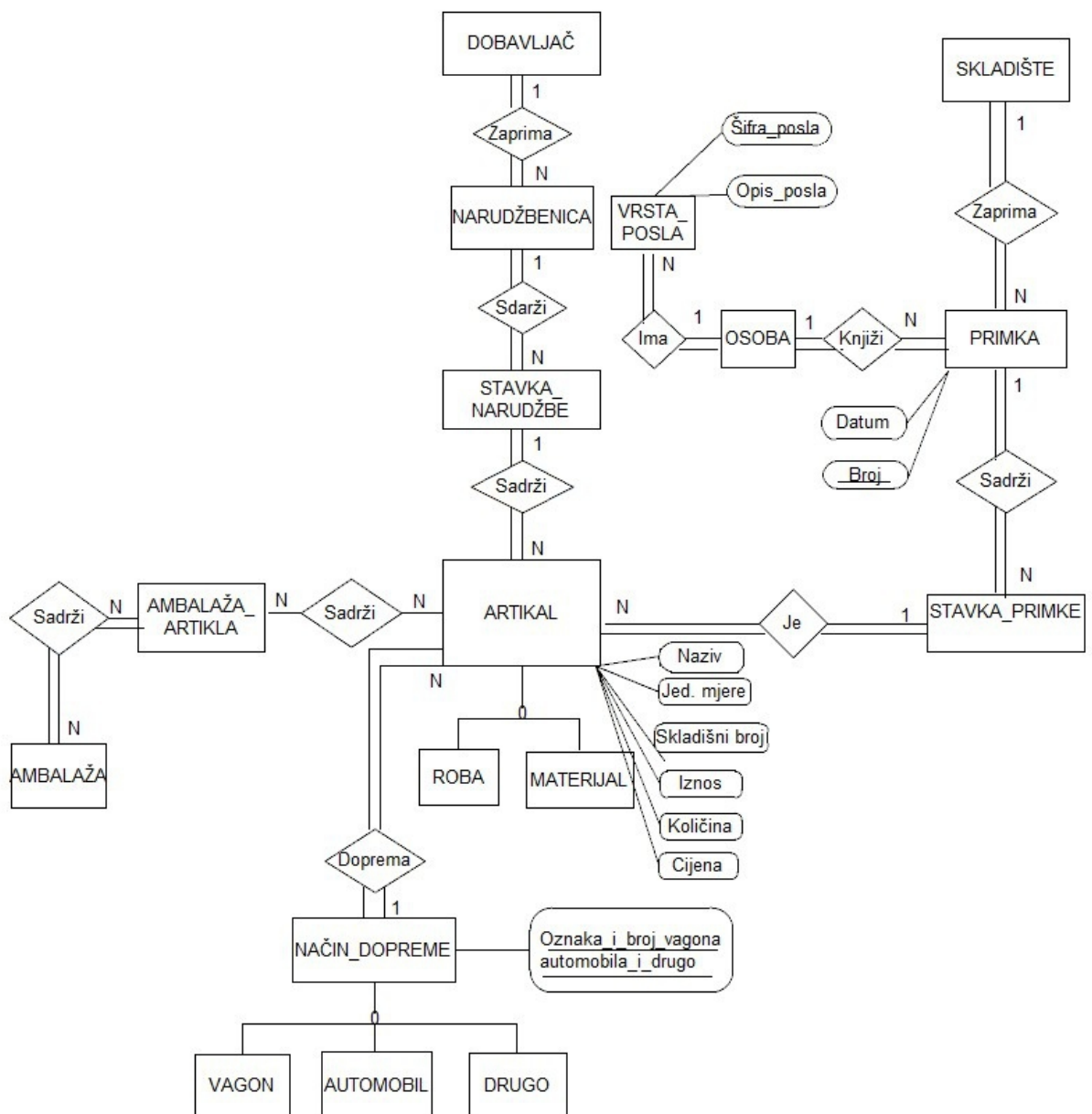
Slika 8: Elmasri & Navatheov prikaz

Izvor: Elmasri, R. & Navathe, S.B., (2011), *Fundamentals of database systems*, 6th ed., USA: Pearson education, p. 223

Slika 8 prikazuje osnovne komponente Elmasri & Navatheovog prikaza.

Dozvoljava n - arne veze. Na ERD razini ne dozvoljava vanjski ključ. Ima četiri vrste atributa: ključni, viševrijednosni, izvedeni i složeni.<sup>49</sup>

<sup>49</sup> Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, pp. 31-32



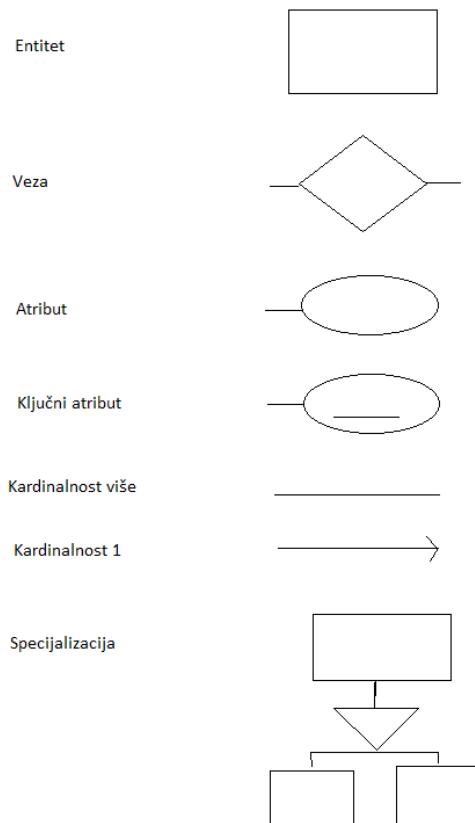
Slika 9: ER dijagram Elmasri & Navathe modelom

Izvor: Izradila studentica

Slika 9 prikazuje ER dijagram na slučaju skladišne primke.

## 6.4. Korth & Silberschatz

Osnovni elementi grafičkog prikaza Korth & Silberschatz su:



Slika 10: Korth & Silberschatzov prikaz

Izvor: Izradila studentica<sup>50</sup>

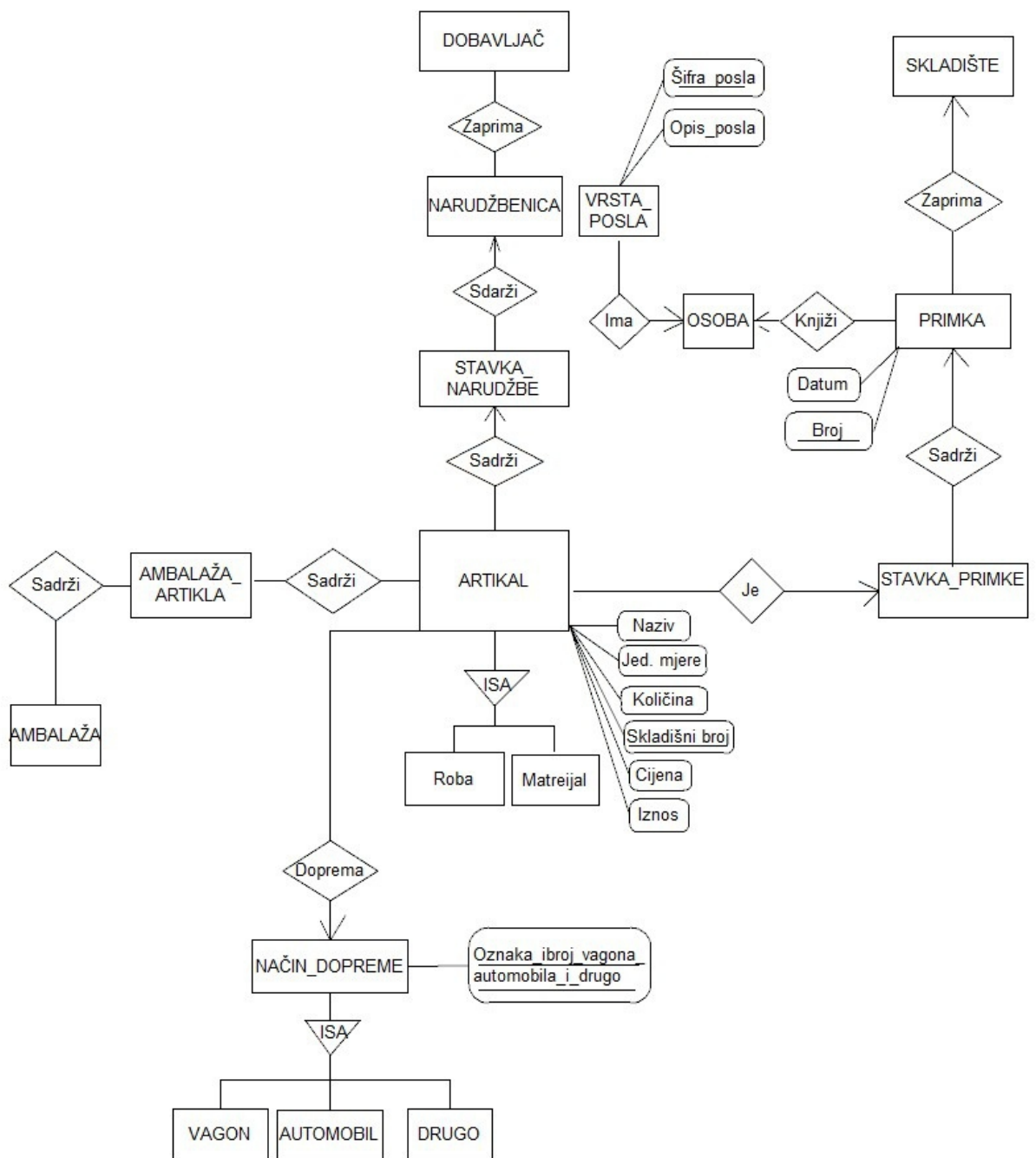
Slika 10 prikazuje osnovne komponente modela Korth & Silberschatz.

Dozvoljava n - arne veze. Na ERD razini ne dozvoljava vanjski ključ.<sup>51</sup>

---

<sup>50</sup> Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, p. 14

<sup>51</sup> Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, pp. 31-32



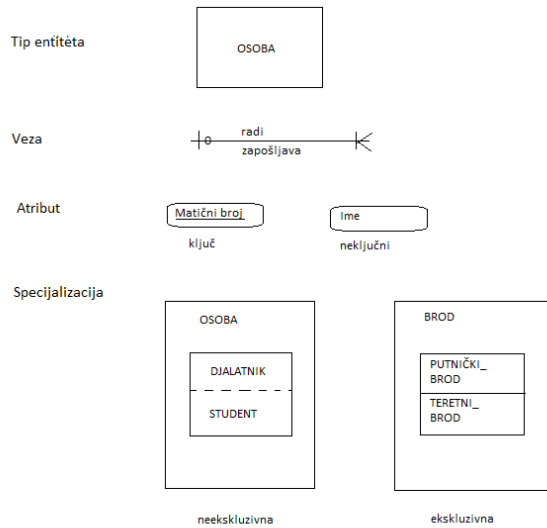
Slika 11: ER dijagram Korth & Silberschatzovim modelom

Izvor: Izradila studentica

Slika 11 prikazuje ER dijagram na slučaju skladišne primke.

## 6.5. Martin

Osnovni elementi Martinovog grafičkog prikaza su:



Slika 12: Martinov prikaz

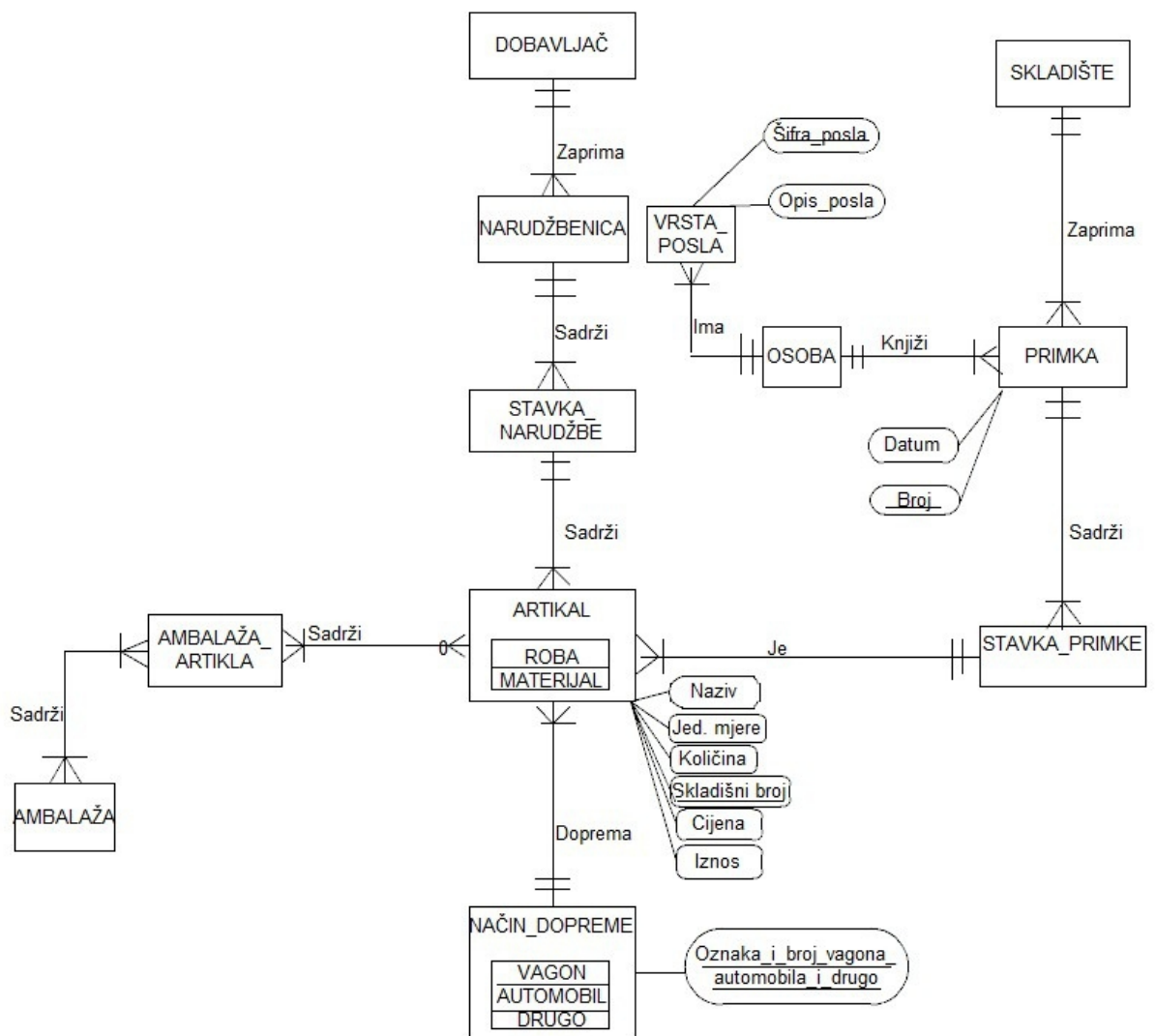
Izvor: Varga, M., (1994), Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka, Zagreb: DRIP, p. 58

Slika 12 prikazuje osnovne komponente Martinovog prikaza.

Dozvoljava binarne veze. Na ERD razini ne dozvoljava vanjski ključ.<sup>52</sup>

---

<sup>52</sup> Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, pp. 31-32

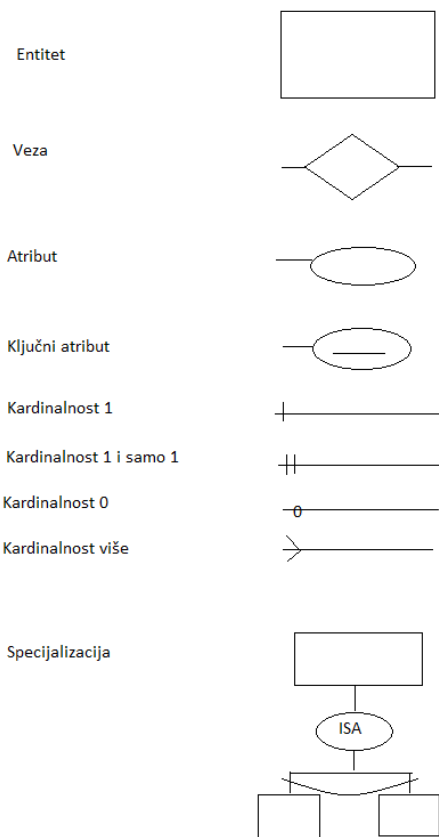


Slika 13: ER dijagram na slučaju primke Martinovim modelom

Izvor: Izradila studentica

Slika 13 prikazuje ER dijagram na slučaju skladišne primke.

## 6.6. Mc Fadden & Hoffer



Slika 14: Mc Fadden & Hofferov prikaz

Izvor: Izradila studentica<sup>53</sup>

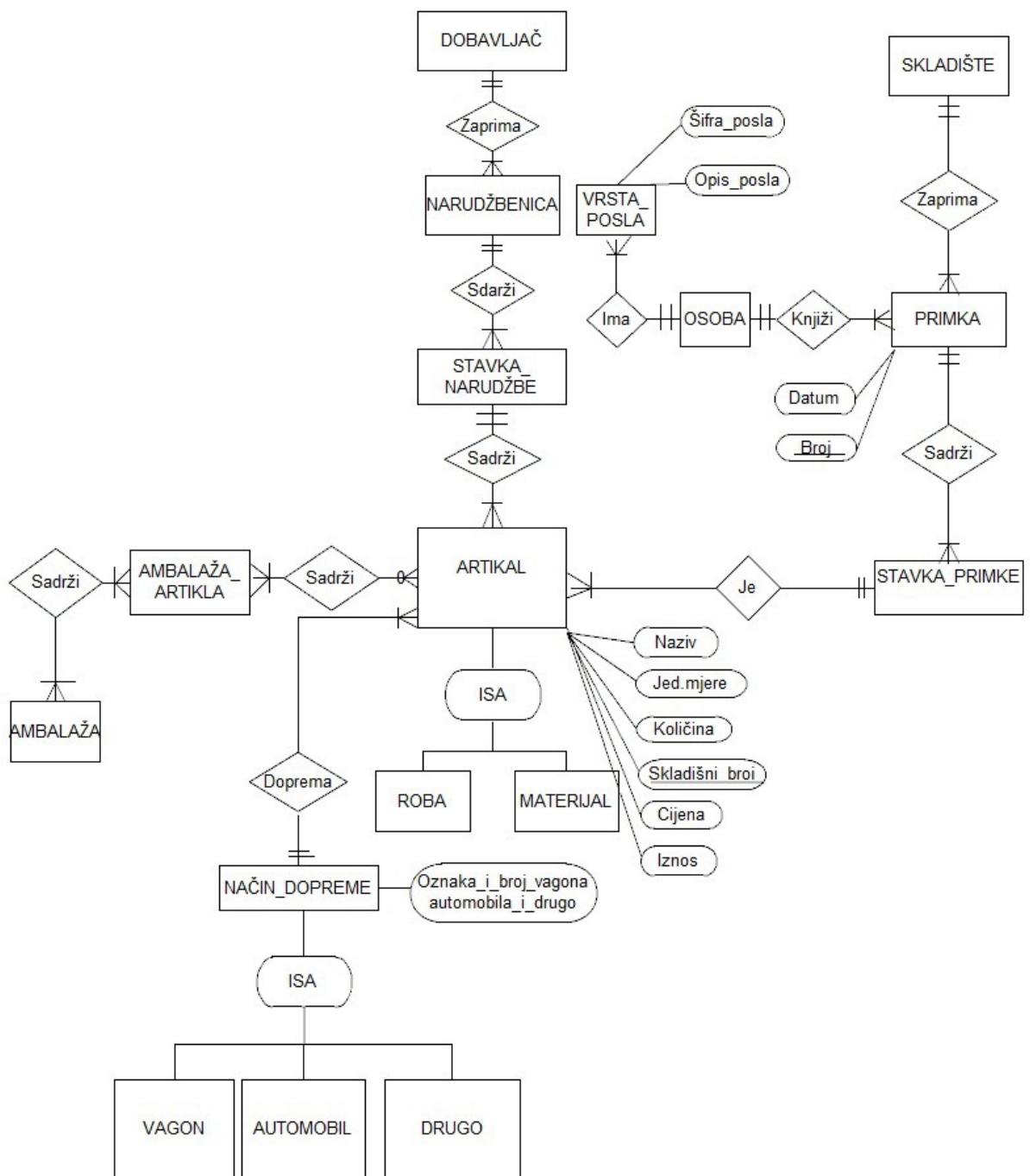
Slika 14 prikazuje osnovne komponente modela Mc Fadden & Hoffer.

Dozvoljava n - arne veze. Na ERD razini ne dozvoljava vanjski ključ.<sup>54</sup>

---

53 Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, pp. 14.-15.

54 Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, pp. 31-32



Slika 15: ER dijagram Mc Fadden & Hoffer modelom

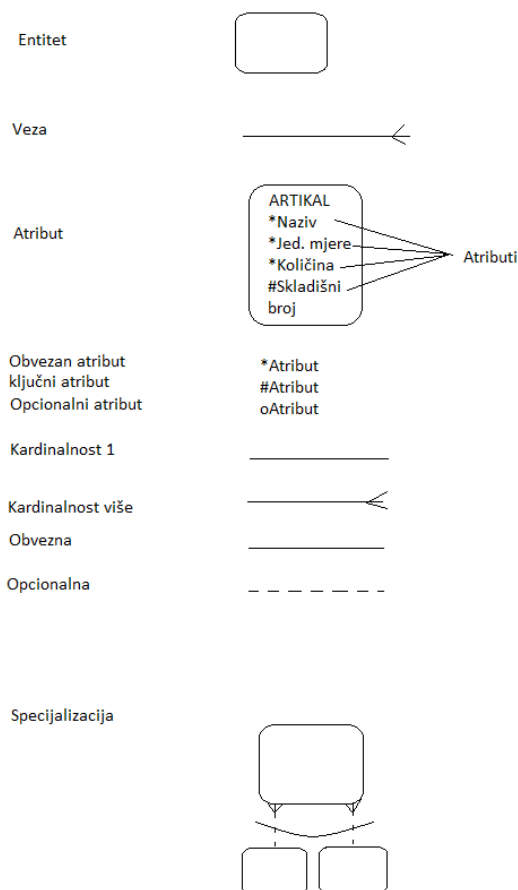
Izvor: Izradila studentica

Slika 15 prikazuje ER dijagram na slučaju skladišne primke.



## 6.7. Oracle's case method notacija (Barkerova notacija)

Osnovni elementi Oracle's case method grafičkog prikaza su:



Slika 16: Oracle's case method prikaz

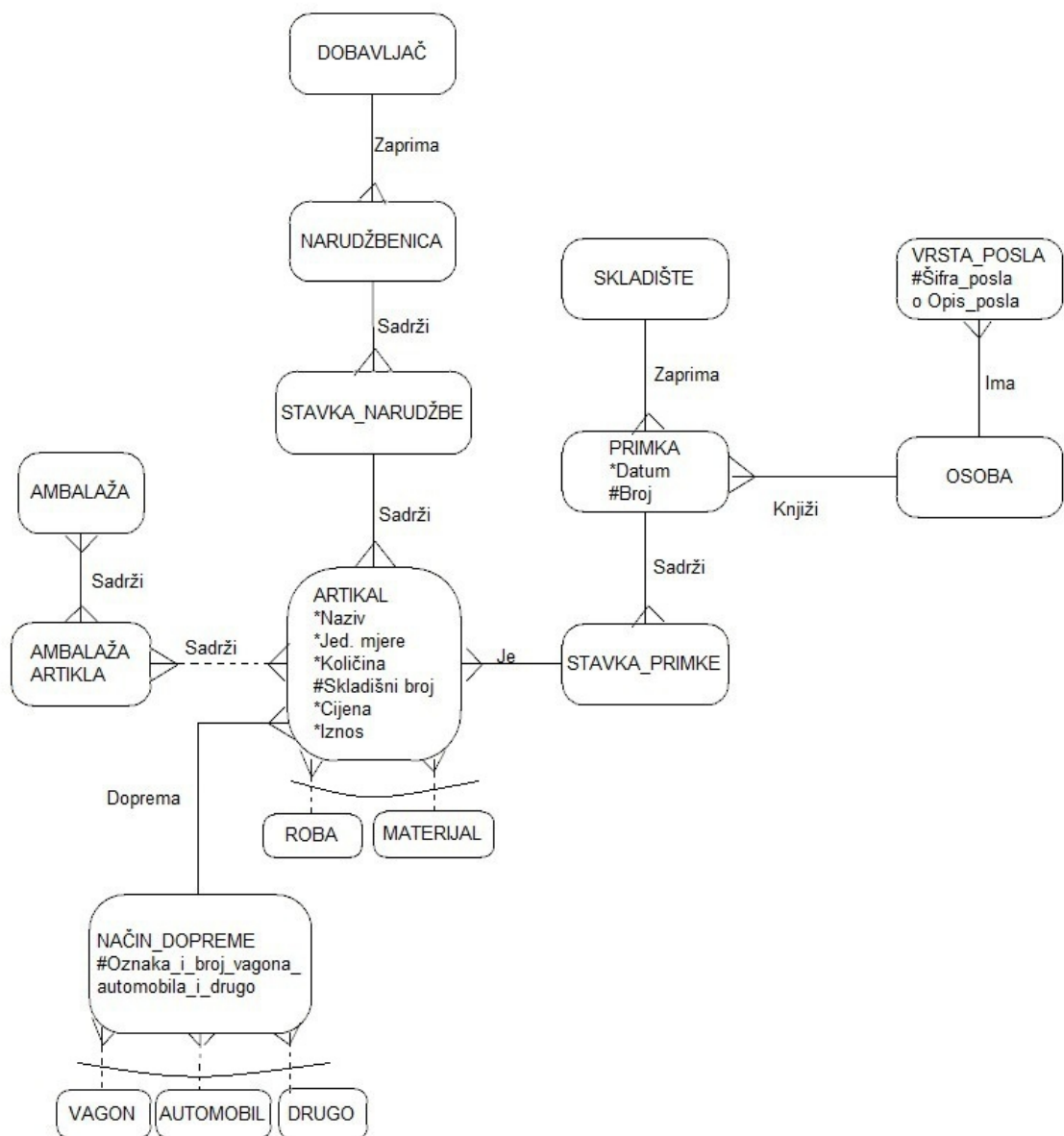
Izvor: Izradila studentica<sup>55</sup>

Slika 16 prikazuje osnovne komponente Oracle's case method prikaza.

Dozvoljava binarne veze. Na ERD razini ne dozvoljava vanjski ključ.<sup>56</sup>

55 Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, pp. 16.-17.

56 Prema: Song, I., Evans, M., Park, E.K., A Comparative Analysis of Entity-Relationship Diagrams, *Journal of Computer and Software Engineering*, Vol.3, pp. 31-32



Slika 17: ER dijagram Oracle's case method modelom

Izvor: Izradila studentica

Slika 17 prikazuje ER dijagram na slučaju skladišne primke.

## 7. Prošireni model entiteta i veza (The enhanced entity - relationship model (EER))

Od kasnih 1970 - ih projektanti baza podataka pokušavali su projektirati preciznije baze podataka koje točnije prikazuju svojstva podataka i ograničenja. To je bilo posebno važno za novu primjenu tehnologije baze podataka kao što su CAD/CAM (engl. computer – aided design/computer aided manufacturing), telekomunikacije, složeni programski sustavi i zemljopisni informacijski sustavi. To je dovelo do razvoja dodatnih koncepata semantičkog modeliranja podataka koji su ugrađeni u konceptualni podatkovni model kao što je model entiteta i veza.<sup>57</sup>

Prošireni model entiteta i veza uključuje sve koncepte modeliranja kao kod modela entiteta i veza i uz to uvodi nove pojmove kao što su: podrazred (engl. subclass), nadrazred (engl. superclass), specijalizacija, generalizacija i nasljeđivanje.<sup>58</sup>

### *Podrazred i nadrazred*

Entitet ZAPOSLENIK opisuje tip svakog entiteta zaposlenika i odnosi se na trenutni skup entiteta ZAPOSLENIK u bazi podataka PODUZEĆE. Entiteti koji su članovi tipa entiteta ZAPOSLENIK mogu se dalje razlikovati u TAJNICA, INŽENJER, MENADŽER itd. Sve ove podgrupe tipa entiteta ZAPOSLENIK zovemo podrazredom entiteta ZAPOSLENIK, a entitet ZAPOSLENIK je nadrazred svakog od ovih podrazreda. Važan koncept povezan sa podrazredima je nasljeđivanje. Entitet koji je član podrazreda nasljeđuje sve attribute i veze u kojima sudjeluje entitet nadrazreda.<sup>59</sup>

### *Specijalizacija*

Specijalizacija je proces određivanja skupa podrazreda entiteta koji se zove nadrazred specijalizacije. Skup podrazreda koji oblikuje specijalizaciju je određen na osnovi nekih različitih karakteristika entiteta u nadrazredu. Npr., skup podrazreda {TAJNICA, INŽENJER,

---

57 Prema: Elmasri, R. & Navathe, S.B., (2011), Fundamentals of database systems, 6th ed., USA: Pearson education , p. 245

58 Ibid., p. 246

59 Ibid., pp. 247-248

TEHNIČAR} je specijalizacija nadrazreda ZAPOSLENIK koji se razlikuje među entitetima ZAPOSLENIK na osnovi vrste posla svakog entiteta ZAPOSLENIK.<sup>60</sup>

### *Generalizacija*

Preokrenuti proces apstrakcije u kojem potiskujemo razlike između više entiteta, prepoznajemo njihove zajedničke značajke i generaliziramo ih u jedan nadrazred od kojeg su prvobitni entiteti podrazredi. Npr., entiteti AUTO i KAMION. Zbog toga što imaju više zajedničkih atributa mogu se generalizirati u entitet VOZILO. AUTO i KAMION su sad podrazredi generaliziranog nadrazreda VOZILO.<sup>61</sup>

### *Ograničenja specijalizacije i generalizacije*

U pojedinim specijalizacijama možemo točno odrediti entitete koji će postati članovima svakog podrazreda tako što ćemo postaviti uvjet na vrijednost od nekih atributa nadrazreda.

Takvi podrazredi nazivaju se predikatno - određeni ( ili uvjetno - određeni) podrazredi. Npr., ako entitet ZAPOSLENIK ima atribut Vrsta\_posla možemo odrediti uvjet članstva u podrazredu TAJNICA pomoću uvjeta ( Vrsta\_posla = „Tajnica“), kojeg zovemo određujući predikat podrazreda. Taj uvjet je ograničenje koje određuje da točno ti entiteti tipa ZAPOSLENIK čija vrijednost za atribut Vrsta\_posla je „Tajnica“ pripada podrazredu. Ako svi podrazredi u specijalizaciji imaju uvjet članstva na istom atributu nadrazreda, specijalizacija se naziva atributno - određena specijalizacija, a atribut se naziva određujućim atributom specijalizacije. Kada nema uvjeta za određivanje članstva u podrazredu, podrazred se naziva korisnički - određenim. Disjunktno ograničenje određuje da podrazred specijalizacije mora biti disjunktan. To znači da entitet može biti član najviše jednog podrazreda specijalizacije.

Cjelovito ograničenje može biti ukupno i djelomično. Ukupno ograničenje specijalizacije određuje da svaki entitet u nadrazredu mora biti član barem jednog podrazreda u specijalizaciji.

---

<sup>60</sup> Ibid., p. 248

<sup>61</sup> Ibid., p. 250

Npr. ako svaki ZAPOSLENIK mora biti ZAPOSLENIK\_PLAĆEN\_PO\_SATU ili ZAPOSLENIK\_S\_PLAĆOM onda je specijalizacija {ZAPOSLENIK\_PLAĆEN\_PO\_SATU, ZAPOSLENIK\_S\_PLAĆOM} ukupna specijalizacija entiteta zaposlenik. Djelomična specijalizacija je npr. ako neki entitet ZAPOSLENIK ne pripada niti jednom podrazredu {TAJNICA, INŽENJER, TEHNIČAR}.<sup>62</sup>

### *Hijerarhije Specijalizacije i generalizacije i mreže*

Podrazred može imati daljnje podrazrede određene oblikujući hijerarhiju ili mrežu specijalizacije. Npr., INŽENJER je podrazred entiteta ZAPOSLENIK i nadrazred entiteta MENADŽER\_INŽENJERSTVA; to predstavlja ograničenje iz stvarnog svijeta prema kojem svaki MENADŽER\_INŽENJERSTVA mora biti INŽENJER. Hijerarhija specijalizacije ima ograničenje da svaki podrazred sudjeluje kao podrazred u samo jednoj podrazrednoj vezi, tj. svaki podrazred ima samo jednog prethodnika, što rezultira u strukturi stabla, odnosno strogoj hijerarhiji. Suprotnost je specijalizacijska mreža gdje podrazred može biti podrazred u više od jedne podrazredne veze.<sup>63</sup>

### *Nasljeđivanje*

Važan koncept povezan sa podrazredima je nasljeđivanje. Tip entiteta je definiran atributima koje posjeduje i tipovima veza u kojima sudjeluje. Zbog toga što entitet u podrazredu predstavlja isti entitet iz stvarnog svijeta iz nadrazreda, morao bi posjedovati vrijednosti za njegove specifične atribute kao i vrijednosti atributa kao član nadrazreda. Kažemo da entitet koji je član podrazreda nasljeđuje sve atribute entiteta koji je član nadrazreda. Entitet također i nasljeđuje sve veze u kojima nadrazred sudjeluje<sup>64</sup>.

Baza podataka sveučilišta prikazana na slici 18 bilježi podatke o studentima i njihovim glavnim predmetima, prijepisima i registraciji, kao i o sveučilišnim smjerovima, sponzoriranim istraživačkim projektima i diplomiranim studentima. Za svaku OSOBU baza podataka sadrži informacije o Imenu, JMBG, Adresi, Spolu i Datumu\_rođenja. Identificirana

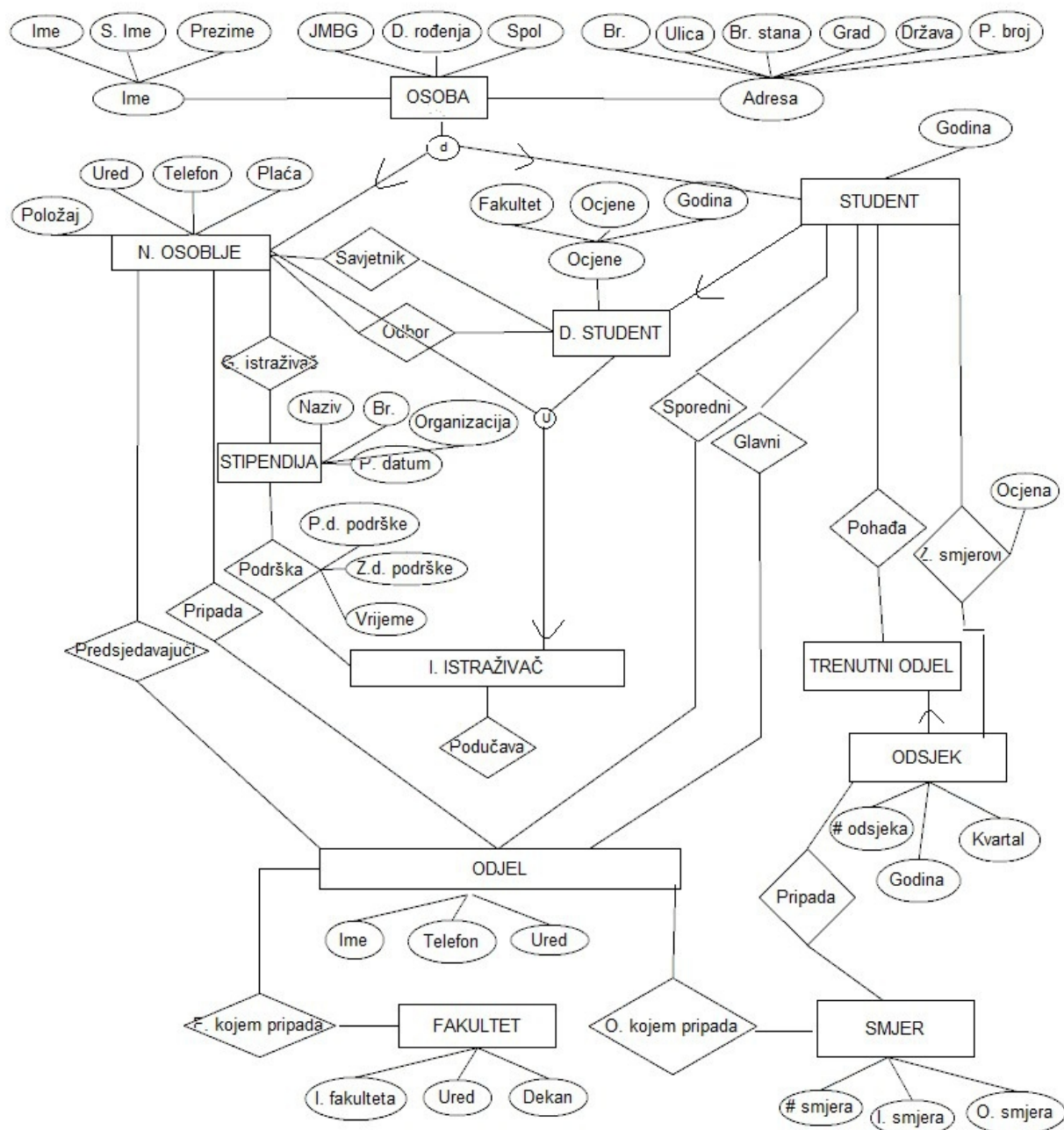
---

62 Ibid., pp. 252-253

63 Ibid., pp. 254-255

64 Ibid., pp. 254-255

su dva podrazreda entiteta OSOBA: NASTAVNIČKO\_OSoblJE i STUDENT. Specifični atributi NASTAVNIČKOG\_OSoblJA su: Položaj, Ured, Telefonski broj ureda i Plaća. Svi članovi fakulteta su povezani s akademskim odjelom, odnosno odjelima, s kojima su povezani (veza PRIPADA).



Slika 18: EER dijagram za bazu podataka sveučilišta

Izvor: Elmasri, R. & Navathe, S.B., (2011), Fundamentals of database systems, 6th ed., USA: Pearson education, p. 262

Specifični atribut studenta je Godina. Svaki student je povezan sa svojim GLAVNIM i SPOREDNIIM odjelima sa sekcijama smjera koje trenutno pohađa i sa završenim smjerovima. Svaka instanca PRIJEPISA sadrži Ocjenu (Grade) koju je student dobio u sekciji smjera. DIPLOMIRANI\_STUDENT je podrazred STUDENTA s određujućim predikatom Godina=5. Za svakog studenta se čuva popis prijašnjih ocjena u složenom, viševrijednosnom atributu Ocjene. Također, povezujemo DIPLOMIRANOG\_STUDENTA sa SAVJETNIKOM i s ODBOROM\_ZA\_DIPLOMSKI\_RAD, ako postoji.

Akademski ODJEL ima attribute Ime, telefon i broj ureda te je povezan s članom fakulteta koji je PREDSDJEDAVAJUĆI i sa fakultetom kojem pripada. Svaki fakultet ima attribute Ime\_fakulteta, broj ureda i ime dekana. Smjer ima attribute: Broj\_smjera, Ime\_smjera i opis smjera. Ponuđeno je više odjela svakog smjera. Svaki odjel ima attribute: Broj\_\_odjela, te Godina i Kvartal. Brojevi odjela jedinstveno identificiraju svaki odjel. Odjeli ponuđeni tijekom kvartala nalaze se u podrazredu TRENUTNI ODJEL od ODSJEKA. Svaki ODSJEK je povezan s INSTRUKTOROM koji je podučavao ili podučava. Kategorija INSTRUKTOR\_ISTRAŽIVAČ je podskup unije NASTAVNIČKO\_OSOBLJE i DIPLOMIRANI\_STUDENT. Entitet STIPENDIJA prati podatke o istraživačkim stipendijama i ugovorima dodijeljenim sveučilištu. STIPENDIJA ima attribute Naziv\_stipendije, Broj\_stipendije, Organizacija\_koja\_je\_dodijelila\_stipendiju i Početni\_datum. STIPENDIJA je povezana sa jednim glavnim istraživačem i sa svim istraživačima koje podržava. Svaka instanca PODRŠKE ima attribute Početni\_datum\_podrške, Završni\_datum\_podrške i Postotak\_vremena\_koje\_će\_biti\_utrošeno\_na\_projektu.<sup>65</sup>

---

65 Prema: Elmasri, R. & Navathe, S.B., (2011), Fundamentals of database systems, 6th ed., USA: Pearson education, pp. 261-263

## Zaključak

Baze podataka omogućavaju lakše stvaranje i održavanje kompleksnih informatičkih sustava. Kako su baze podataka preslika stvarnosti, u procesu njihova stvaranja potrebno je pravilno modelirati poslovne procese i zahtjeve baze kako bi bila prilagođena organizaciji kojoj je namijenjena. Tu je od značajne uloge modeliranje entiteta i veza, odnosno dijagrami koji olakšavaju komunikaciju između projektanta i korisnika. ER modeliranje se radi kako bi se opisali korisnički zahtjevi. Model entiteta i veza kao dio konceptualnog modeliranja zadržava samo one potrebne detalje (entitete, veze i njihove atribute) i izbacuje suvišne. Iz analize različitih ER - modela možemo zaključiti kako Elsmari & Navathe ima najkompleksniji prikaz koji nam daje najbolji uvid u određeni poslovni slučaj (Dozvoljava n - arne veze. Na ERD razini ne dozvoljava vanjski ključ. Ima četiri vrste atributa: ključni, viševrijednosni, izvedeni i složeni.).



## LITERATURA

6. de Haan, L. et al., (2009), *Beginning Oracle SQL*, [CD], US: Apress
7. Manger, R., (2012), *Baze podataka*, Zagreb: Element d.o.o
8. Song, I., Evans, M., Park, E.K., *A Comparative Analysis of Entity - Relationship Diagrams*, *Journal of Computer and Software Engineering*, Vol.3
9. Radovan, M, (1993), *Baza podataka: Relacijski pristup i SQL*, Zagreb: Informator
10. Ramakrishnan, R. & Gehrke, J., (2003), *Database management systems*, 3rd ed., New York: McGraw - Hill
11. Varga, M., (1994), *Baze podataka: Konceptualno, logičko i fizičko modeliranje podataka*, Zagreb: DRIP
12. Varga, M., (2014), *Upravljanje podacima*, 2.izdanje,Zagreb: Element d.o.o.,
13. Anonymous, (n.d.),*Information Systems: Introduction and Concepts*,[Internet], <raspoloživo na:

[https://mitpress.mit.edu/sites/default/files/titles/content/9780262015387\\_sch\\_0001.pdf](https://mitpress.mit.edu/sites/default/files/titles/content/9780262015387_sch_0001.pdf)>,

[pristupljeno 28.02.2016.]

## POPIS SLIKA

Slika 1. Model životnog ciklusa informacijskog sustava poduzeća

Slika 2. Glavne faze oblikovanja baze podataka.....13

## Sažetak

Baza podataka je skup podataka. Arhitektura baze podataka sastoji se od tri sloja: Fizička razina, globalna logička razina i lokalna logička razina.

Prednosti korištenja baze podataka su kontrola redundance, integralnost sustava, korištenje zajedničkih podataka, zaštita podataka, standardizacija podataka i optimizacija cjeline. Modeli baze podataka su relacijski model, mrežni model, hijerarhijski model i objektni model. Glavne faze oblikovanja baze podataka su prikupljanje i analiza zahtjeva, konceptualno modeliranje, logičko modeliranje i fizičko modeliranje. Najupotrebljiviji konceptualni model je model entiteta i veza. Chen je u svom članku 1977. predstavio model koji pomoću dijagrama prikazuje stvarni svijet. E - R model koristi prirodni pogled na svijet koji se sastoji od entiteta i veza. Entitet predstavlja stvarnu ili apstraktnu pojavu koja se može identificirati. Veze su odnosi među entitetima. Atributi su svojstva kojima je opisan entitet. Ključ jednoznačno identificira entitet. Koraci u otkrivanju entiteta, veza i atributa: 1. Otkrivanje imenica (entiteti i atributi) i glagola (veze), 2. Odrediti koji atributi opisuju koji entitet, 3. crtanje dijagrama i 4. pisanje popratnog teksta. Na temelju skladišne primke

prikazali smo neke od E - R modela, a to su: Batini, Ceri & Navathe, Chen, Elmasri & Navathe, Korth & Silberschatz, Martin, McFadden & Hoffer i Oracle's case method (Barker). Prošireni model entiteta i veza razvijao se od kasnih 1970 - ih kako bi se omogućilo postizanje zahtjevnijih i preciznijih baza podataka. Prošireni model entiteta i veza uvodi nove pojmove kao što su podrazred, nadrazred, specijalizacija, generalizacija i nasljeđivanje.

Ključne riječi: Baze podataka, entitet, atribut, veza, ključ, prošireni model entiteta i veza

## Summary

Database is a collection of interconnected data that constitutes a common data base on all applications of an information system. Architecture of a database consists of three layers: Physical, global logical level and a local logic level.

The advantages of using a database are redundancy control, system integrity, the use of common data, data protection, data standardization and optimization of the unit. Models of databases are relational model, network model, the hierarchical model and the object model. The main stages of database design are requirements gathering and analysis, conceptual modeling, logical modeling and physical modeling. The most used conceptual model is a model of entities and relationships. Chen in his article "The Entity - relationship model: TOWARD A UNIFIED VIEW OF DATA" in 1977 introduced a model with diagrams showing the real world. E - R model uses a more natural view of the world consisting of entities and relationships.

The entity is real or abstract phenomenon that can be identified. Relationships are relations between the entities. Attributes are properties which are describing an entity. The key uniquely identifies the entity. Steps in detecting entities, relationships and attributes: 1. Detection of nouns (entities and attributes) and a verb (relationships), 2. Determine what attributes describe the entity, 3. drawing diagrams and 4. writing the accompanying text. Based on a warehouse receipt, we presented some of the ER model, and those are: Batini, Ceri & Navathe, Chen, Elmasri & Navathe, Korth & Silberschatz, Martin, McFadden & Hoffer and Oracle's case method (Barker).

The extended model of entities and relationships was developed from the late 1970s to enable the achievement of demanding and precise database. The extended model of entities and relationships introduces new concepts such as class, superclass, specialization, generalization and inheritance.

Key words: Database, entity, attribute, relationship, the extended model of entities and relationships.