

BAZE PODATAKA

SKRIPTA

1.1 Pojam podatka

- **Podatak** je iskaz definisan jednom prostom izjavnom rečenicom
- **Informacija** je novi podatak koji poseduje neku relevantnu novinu, a rezultat je obrade poznatih podataka.

1.2 Entitet

- **Entitet** je digitalna slika, opisana ograničenim brojem diskretnih vrednosti (atributima), nekog kontinualnog procesa koji se odvija u realnom svetu (nekog dela realnog sveta).
- Svojstva entiteta opisuju se atributima.

1.3 Model podatka

- **Model podataka** služi za prikazivanje objekata sistema, njihovih atributa i njihovih međusobnih veza.
- Danas, za formiranje modela realnog sveta dominantan je model objekat-veze (MOV) ili Entity-Relationship model (E-R model). Relacioni model se najčešće koristi za implementaciju.
- U zavisnosti od karakteristika tabela, atributa a naročito veza među tabelama razlikujemo 3 osnovna modela projektovanja: **hijerarhiski, mrežni i relacioni model**.
- Svaki model mora imati mogućnost: definisanja podataka, definisanja pravila za očuvanje podataka, definisanje pravila manipulacije podacima.

1.3.1 Hijerarhiski model

- Sve veze među tabelama (atributima) moraju biti precizno određene.
- Zahteva stručnost svih korisnika (ne samo administratora)
- Ne kontrolisan gubitak informacija (brisanjem „roditelja“ gube se i tabele „naslednici“)
- Redundanca podataka – potreba za memorisanjem istih podataka na više mesta.

1.3.2 Mrežni model

- U odnosu na hierarhiski model omogućena je veza tipa M:N. (jedan naslednik može imati više roditelja.

1.3.3 Relacioni model

- Najpopularniji model, danas najdominantniji.
- Struktura veoma jednostavna, baza predstavlja skup tabela.
- Moguća formalno matematička interpretacija tabela.
- Pošto I relacija(veza) između tabela može da se predstavi kao tabela, u relacionom modelu sve je definivano samo tabelama.
- Relaciona baza podataka se sastoji od vremenski promenljivih tabela (relacija) koje mogu biti bazne i izvedene.

- Baze se još nazivaju fizičke baze zato što su zapisane u memoriji.
- Izvedene se izvode iz fizičkih upitima.
- U relacionim bazama postoji ne zavisnost aplikacije od podataka tako da se podaci mogu obrađivati pomoću raznih aplikacija što ih čini dostupnijim.
- SUBP – sistemi za upravljanje bazama podataka moraju imati razvijene sisteme za zaštitu podataka.

Glava 2. Relacioni model

2.1 Pojam relacije

- Kompletan zasnovan na relacionoj algebri – omogućen razvoj specifičnog softvera i obrada uz garantovanu konzistentnost podataka.
- U relacionim bazama podataka pojam relacija je u stvari sinonim za tabelu uz određena ograničenja.
 - Među atributima tabele(relacije) ne sme postojati funkcionalna zavisnost
 - Unutar jedne relacije ne smeju postojati dve identične n-torke
 - Redosled n-torki u relaciji je ne bitan
 - Svi atributi unutar jedne tabele moraju imati različita imena

2.2 Entitet je deo realnog sveta opisan ograničenim brojem atributa.

2.3 Atribut je jedno od svojstava entiteta.

2.4 Domen je skup vrednosti koje atribut može da poprими.

2.5 Primarni ključ je atribut (ili više njih kada je to potrebno) čija vrednost jednoznačno određuje samo jednu n-torku u tabeli. Dve n-torke ne smeju imati istu vrednos ključa.

2.6 Spoljni ključ je običan atribut u jednoj tabeli koji je u drugoj tabeli primarni ključ. Koristi se za uspostavljanje veze u tabelama.

2.7 Coddova pravila

- E.F. Codd je u svojim radovima definisao 12 pravila, od kojih najmanje 6 mora biti ispunjeno da bi se informacioni sistem smatrao relacionim.
 0. Svaki sistem za upravljanje bazom podataka koji se smatra relacionim mora imati mogućnost upravljanja bazom podataka na relacioni način i relacionim metodama.
 1. Sve informacije u Relacionim bazama moraju logički biti predstavljene na isti način (vrednostima podataka u tabelama)
 2. Svaki podatak korisniku mora biti dostupan uz pomoć imena relacije u kojoj se nalazi, vrednosti primarnog ključa i imena atributa.
 3. Mogućnost prikazivanja nepostojeće informacije (null)
 4. Dinamički katalog – autorizovanim korisnicima mora biti omogućeno da primenom nekog relacionog jezika menjaju bazu.
 5. Softverski paketi za obradu (manipulaciju) data bazom mora, precizno definisanom sintaksom i na korisniku blizak način omogućiti: definisanje tabele, definisanje atributa, unošenje podataka, definisanje proizvoljnog „pogleda“ na bazu podataka, manipulaciju podacima, postavljanje ograničenja korisnicima, autorizaciju korisnika, upravljanje proizvoljnim transakcijama.

6. Nezavisnost integriteta baze.
 - Integritet entiteta: nijedan atribut koji je primarni ključ ili deo primarnog ključa neke bazne relacije ne može da uzme null vrednost. (primarnog ključa nema da nema!)
 - Referencijalni integritet: skup vrednosti spoljnog ključa relacije R1 mora biti podskup skupa vrednosti primarnog ključa relacije R2 sa kojom se povezuje R1. Relacije R1 i R2 ne moraju biti različite.
 7. Jezikom treće generacije se ne mogu zaobići pravila integriteta definisana relacionim jezikom.
 8. Mogućnost kreiranja pogleda i definisanja operacija nad njima.
 9. Nad pogledima se mogu izvršavati i operacije održavanje baze podataka.
 10. Aplikacioni programi ostaju ne promenjeni ako se promeni fizička organizacija baze ili fizični metod pristupa.
 11. Aplikacioni programi ostaju ne promenjeni ako se promene bazne tabele (sem u promeni struktura tabela).
 12. Sve navedene karakteristike su ne zavisne od distribucije baze podataka.
- 2.8 Distribuirane baze podataka
- Pod distribuiranom bazom podataka smatra se danas svaki informacioni sistem koji radi u računarskoj mreži, i koji ima najmanje dva dislocirana računara, sa dislociranim tabelama u kojima se nalaze podaci.

Glava 3 Osnove relacione algebre

- Da bi se mogao shvatiti način rada relacionih baza podataka potrebna su osnovna znanja iz relacione algebre.
 - Osnovni elementi relacione algebre su:
 - osnovni operatori
 - operatori pridruživanja (kojima se relacije povezuju)
 - E.F. Codd definisao je dve grupe operatora
 - tradicionalni operatori
 - specijalni operatori
- 3.1 Tradicionalni operatori su pogodni za ažuriranje i izvode se nad bar 2 relacije. A to su:
- unija (union)
 - presek (intersect)
 - razlika (difference)
 - proizvod (cartesian product)

Tabela A

Šifra#	Prezime	Ime	Tel broj
--------	---------	-----	----------

3244	Aksentijević	Petar	071 334 952
1772	Maksimović	Ilija	015 723 543

Tabela B

Šifra#	Prezime	Ime	Tel broj
3244	Aksentijević	Petar	071 334 952
2345	Petrović	Dara	081 17318

3.1.1 Unija

- Unija dve relacije (tabele) je tabela koja se sastoji od svih elemenata koji pripadaju datim tabelama.
- Da bi rezultat bio relacija moraju se zadovoljiti sledeća pravila:
 - obe relacije moraju imati iste atribute
 - isti atributi moraju biti definisani nad istim domenom

Primer:

Tabela A **U** Tabela B

Šifra#	Prezime	Ime	Tel broj
3244	Aksentijević	Petar	071 334 952
1772	Maksimović	Ilija	015 723 543
2345	Petrović	Dara	081 17318

3.1.2 Presek

- Presek dve relacije je nova relacija koja sadrži sve n-torke koje su zajedničke za obe relacije.

Primer:

Tabela A Tabela B

Šifra#	Prezime	Ime	Tel broj
3244	Aksentijević	Petar	071 334 952

3.1.3 Razlika

- Razlika dveju kompatibilnih relacija je nova relacija koja ima iste atribute kao te relacije, a telo joj se sastoji od onih n-torki koje se nalaze u relaciji A a ne nalaze se u relaciji B.
- U relacionoj algebri razlika se označava A/B
- $A-B \neq B-A$

Primer:

A-B

Šifra#	Prezime	Ime	Tel broj
1772	Maksimović	Ilija	015 723 543

B-A

Šifra#	Prezime	Ime	Tel broj
--------	---------	-----	----------

2345	Petrović	Dara	081 17318
------	----------	------	-----------

3.1.4 Proizvod

- Proizvod u relacionoj algebra se definiše kao dekartov proizvod gde su elementi množenja cele n-torke.

Alfa	
A	B
a1	b1
a2	b2
a3	b3

*

Beta		
C	D	E
c1	d1	e1
c2	d2	e2

=

A	B	C	D	E
a1	b1	c1	d1	e1
a1	b1	c2	d2	e2
a2	b2	c1	d1	e1
a2	b2	c2	d2	e2
a3	b3	c1	d1	e1
a3	b3	c2	d2	e2

3.2 Specijalni operatori pogodni za izveštavanje

3.2.1 Selekcija (ograničenje,restrikcija)

- Pravi novu relaciju u koju se smeštaju samo one n-torke koje zadovoljavaju uslov koji je predhodno definisan.

3.2.2 Projekcija

- Pravi novu relaciju koja sadrži samo izabrane attribute originalne relacije.

3.2.3 Spajanje

- Postoji mnogo vrsta spajanja od kojih su najvažnije:
 - Prirodno spajanje
 - Spajanje pod nekim uslovom
- Prirodno spajanje relacija A I B daje novu relaciju AB u kojoj se nalaze svi atributi relacije A i atributi relacije B kojih nema u relaciji A.
- Spajanje pod uslovom se izvodi kada dve relacije nemaju ni jedan zajednički atribut. Rezultat spajanja je običan kartezijski proizvod.

3.2.4 Operacija deljenja

- Da bi A:B bilo izvodljivo, potrebno je da se svi atributi relacije B nalaze u relaciji A.
- Rezultat deljenja je relacija C koja ima samo one attribute relacije A koje ne sadrži relacija B i one n-torke koje sadrže vrednosti atributa relacije B.

Primer:

A	
X#	Y#
017	a22
033	a43
077	a86
061	a43
044	a43

:

B
Y#
a43

=

C
X#
033
061
044

3.3 Dodatni operatori

- U praksi se pokazalo da osnovni operatori ne mogu da zadovolje sve zahteve tako da su uvedeni još i operatori
 - proširenja
 - agregacije
 - uopštenog deljenja
 - spoljnog spajanja
 - uslovni operator (maybe)

Glava 5 Sinteza relacionog modela

- Sinteza relacionog modela podrazumeva izradu logičkog modela baze podataka u šta spada:
 - analiza podataka (postojećih, potrebnih)
 - definisanje relacija (tabela) i veza među njima
 - dovođenje modela na relacioni oblik
- Logički model baze – struktura i oblik baze
- Sam fizički skup podataka – sadržaj baze
- Do logičkog modela baze može se doći
 - sintezom modela objekat-veze (MOV, E-R model)
 - postupkom normalizacije tabela sa podacima mada se često kombinuju.

5.1 E-R model (MOV)

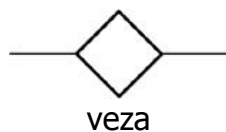
- Najkraća definicija: dobijanje saznanja o objektima, vezama među njima, te svojstvima objekta i veza. Ma šta to značilo !?!?!?!?
- Uspešno sintetizovan E-R model prevodi se na relacioni model.

5.1.1 Osnovne definicije i pojmovi E-R modela

- U svom radu P.P. Chen je uveo nekoliko novih pojmova koji u suštini ne menjaju osnovni Coddov koncept ali ga čine dostupnijim.
- Objekte određene atributima Chen je podelio na:
 - objekte, u užem smislu te reči
 - veze ili vezne objekte

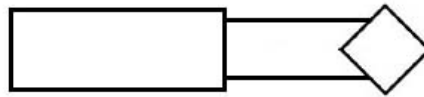
5.2.2 Objekti

- U E-R modelu objektom se smatra onaj entitet koji je pored primarnog ključa opisan još nekim atributom.
- Atribut je svojstvo objekta koje se može opisati jednim podatkom.
- Da bi razlikovali vezni objekat od „običnog“ objekta, grafički ga predstavljamo kao romb koji je sa strane veze N (više) obojen crno a sa strane 1 belo. Opcionalnost veze označavamo kržićem.
- Objekte predstavljamo kao blokove blok-šeme.



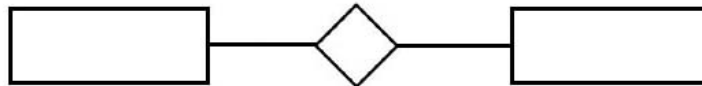
- Pored podela na vezne i stvarne objekte postoji i podela na
 - čvrste objekte

- slabe objekte, koji na neki način (egzistencijalno ili indentifikacijski) zavise od jednog ili više čvrstih objekata.
 - Slabi objekti moraju imati složen primarni ključ koji se sastoji od ključa slabog objekta i ključa čvrstog kome pripadaju.
- 5.1.2 Veze ili vezni objekti
- Svaka veza se karakteriše sa tri osobine: red veze, način uspostavljanja veze i tip veze.
 - **Red veze** određuje broj objekata koji čine samu vezu, a najčešće su:
 - unarne veze
 - binarne veze
 - trojne veze
 - Takođe veze mogu biti i višeg rade (n-tarne)
 - Unarne ili unutrašnje veze se uspostavljaju unutar jednog objekta (tabele) i relativno su retke.



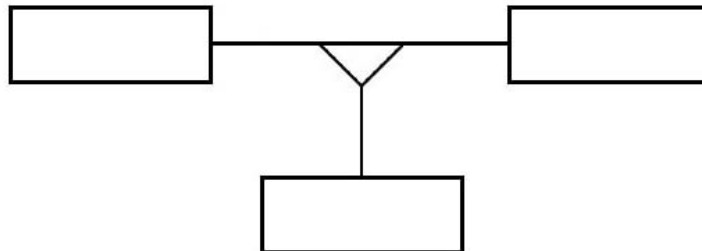
grafičko predstavljanje

- Binarna veza je veza između dva objekta. Najčešće se sreće u praksi.



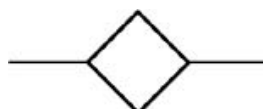
grafičko predstavljanje

- Trojna veza nastaje obično daljim proširenjem binarne veze.

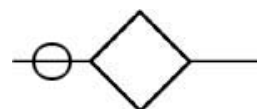


grafičko predstavljanje

- Veze većeg reda od tri se iskazuju na sličan način kao i trojne.
- Treba ih izbegavati pošto su ne pregledne i teške za analizu.
- **Način uspostavljanja veze** može biti
 - obavezan
 - opcionalan
- Opcionalna veza se grafički obeležava kružićem sa strane veze (na kojoj je ona opcionalna).



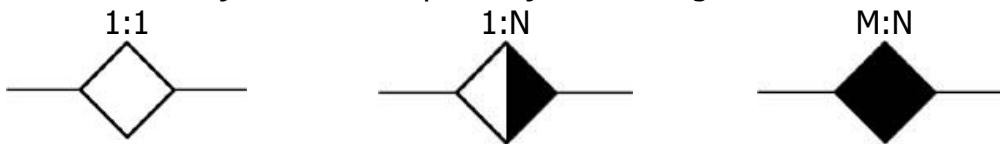
obavezna veza



opcionalna veza

- **Tip povezanosti** može da bude

- tip jedan
- tip mnogo
- Kombinovanjem ova dva tipa dobijamo tri moguća oblika veze:



5.2 Prevođenje E-R model na relacioni oblik

- Tehnika prevođenja E-R model na relacioni oblik izvodi se tako što:
 - svaki objekat postaje relacija
 - svaka veza, ali ne mora, da postane vezna relacija
 - ime objekta postaje ime relacije
 - karakteristike objekta postaju njegovi atributi
 - identifikatori objekta postaju ključevi relacija
- Glavni problem u ovom postupku je prevođenje veza na relacionu formu, jer ih ima 3 vrste, a veze tipa M:N moraju biti razložene jer ih relacioni model ne podržava.

5.2.1 Prevođenje binarnih veza na relacioni oblik

- Binarne veze tipa 1:1 prevode se na relacioni oblik tako što se u jednu tabelu, koja učestvuje u vezi, uvrsti primarni ključ druge tabele kao atribut. Veza se prema tome iskazuje spoljnjim ključem.
- Veze tipa 1:1 po pravilu nemaju svojih atributa jer se svaki eventualni atribut može pripisati jednom od objekata.
- Prilikom izbora tabele u koju treba da doamo spoljnji ključ treba samo obratiti pažnju da ta tabela ima što manje null vrednosti.
- Binarne veze tipa 1:N prevode se na relacioni oblik slično kao veze 1:1. Veza se takođe iskazuje spoljnjim ključem ali ne u bilo kojoj tabeli, već onoj koja je u vezi E-R modela bila na strani N (više).
- Binarne veze tipa M:N prevode se na relacioni oblik obaveznom uvođenjem nove relacije (vezne relacije) u bazu podataka. Ključ te nove relacije je po pravilu složen i sastoji se od primarnih ključeva objekata koji učestvuju u vezi, a atributi su svojstva veze.

5.2.2 Prevođenje unarnih veza na relacioni oblik

- Unarne veze postoje de facto među n-torkama jedne tabele.
- Unarne veze tipa 1:1 i 1:N prevode se na relacioni oblik uvođenjem šifre jedne n-torke (koja učestuje u vezi) kao spoljnjeg ključa u drugu, koja je u vezi sa njom.
- Ako je veza 1:1 atribut koji ima funkciju spoljnjeg ključa ne sme imati mogućnost ponavljanja, tako da jedna vrednost primarnog ključa može biti dodeljena samo jednoj vrednosti spoljnjeg.
- Unarna veza tipa N:M prevodi se na relacioni oblik uvođenjem nove vezne tabele čiji je ključ složen – sastavljen od primarnog ključa originalne tabele i spoljnjeg ključa.
- Kod veza tipa M:N unutar samog objekta treba obratit pažnju da ne dođe do povratne sprege (da neki proizvod ne postane deo samog sebe) koja može bitno uticati na stabilnost sistema.

5.2.3 Prevođenje veza reda većeg od dva

- N-tarne veze mogu tvoriti sledeće tipove : povezanost svih objekata tipa jedan, povezanost jednog objekta tipa mnogo a ostalih tipa jedan, povezanost dveju objekata tipa mnogo, a ostalih tipa jedan, povezanost svih objekata tipa mnogo.
- N-tarne veze prevode se po pravilu na relacioni oblik uvek uvođenjem dodatnih veznih relacija koje uključuju primarne ključeve objekata koji tvore vezu.

5.3 Normalizacija

- Projektovanju baze podataka može se prići i na drugi način, a ne samo E-R modelom, na prvi pogled i jednostavniji, prostim skupljanjem svih raspoloživih podataka.
- Logički model ovako dobijene baze podataka mora da bude tako organizovan da ne sadrži redundancu da bi se sistem mogao efikasno koristiti (pretraživati, ažurirati, brisati itd.)
- Tabele sa sirovo nabacanim podacima retko kad zadovoljavaju sve uslove kako bi baza funkcionisala nesmetano.
- Optimalan izbor relacija i njihovih atributa naziva se normalizacija.
- Normalizacija je u stvari postupak izmene (najčešće rastavljanje na više relacija) prvokoncipiranih tabela sa ciljem postizanja dobrih osobina modela.
- Dekompozicijom neke relacije (tabele) koja nije normalizovana dobijaju se uvek dve ili više tabela koje su obavezno međusobno povezane, i koje se nalaze u nekoj od normalnih formi.
- Normalizacija može da se upotrebi kao sredstvo za proveru i unapređenje predloženog E-R modela.
- Tehnički postupak normalizacije svodi se na relacione operacije projekcije i spajanja, i to na:
 - na dekompoziciju tabela generisanjem vertikalnih podskupova (operacijom projekcije) i
 - na generisanje novih relacija iz dvaju ili više dobijenih vertikalnih podskupova (operacijom spajanja).
- Ovakav pristup sintezi baze podataka naziva se još i vertikalna normalizacija. U sklopu vertikalne normalizacije definisano je do sada 5 normalnih formi, od kojih su najvažnije druga i treća, na kojima se često normalizacija i zaustavlja.
- Pored vertikalne postoji i horizontalna normalizacija koja nije teoriski dovršena, i vezana je za distribuirane data baze.
- Prisustvo redundance u bazi dovodi do niza problema pri korišćenju a naročito ažuriranju podataka. Ti problemi nazivaju se anomalije pri održavanju. Objasnimo ih na jednom primeru:
- Imamo jednu trgovačku firmu koja hoće da vodi evidenciju o svom poslovanju, tačnije o svojim kupcima i artiklima.
- Nenormalizovana relacija bi izgledala ovako:
Narudžba <sifkup#,ime,adresa,količina,sifart#,naziv,kvalitet,cena>

- Anomalija pri upisu podataka sastoji se u tome da upis podataka o nekom kupcu nije moguć dok on nešto ne kupi. Analogno tome upis novog artikla nije moguć dok on ne bude kupljen.
- Anomalija pri brisanju podataka, ako je samo jedan kupac, kojeg želimo da obrišemo, kupio određeni artikl istovremeno ćemo izgubiti i podatke o tom artiklu.
- Anomalija pri izmeni podataka, ako želimo da promenimo ime ili adresu kupca ili karakteristike artikla, izmenu moramo izvršiti na svim mestima gde je podatak zapisan. Što pored povećanja obima posla, povećava mogućnost ljudske greške.

5.3.1 Prva, druga i treća normalna forma

- Sve anomalije koje redundanca podataka donosi mogu biti eliminisane ako se prvobitna relacija dovede u potrebnu normalnu formu. Čitaj 3NF
- Treba obratiti pažnju da pri postupku normalizacije ne dođe do gubitka podataka.
- **Prva normalna forma (1NF)** Relacija se nalazi u prvoj normalnoj formi ako je njen domen (podaci u tabeli) skup atomiranih vrednosti.
- Kako je ovo uslov da bi tabela uopšte bila relacija, sledi da se svaka tabela nalazi u prvoj normalnoj formi.
- Pojava redundance te mogućnost svih pratećih grešaka ukazuju na to da 1NF nije ni približno upotrebljiva za sintezu upotrebljive baze. Zato je neki autori nazivaju i nenormalizovana forma.
- **Druga normalna forma (2NF)** Relacija se nalazi u drugoj normalnoj formi ako svaki atribut koji nije ključni potpuno (a ne delimično) od ključnog atributa
- 2NF nema većeg praktičnog značaja, pošto su sve anomalije i dalje prisutne.
- Primer: Laborant <šifrab#,ime,prezime,brojsobe,telefon>
 - Relacija laborant se nalazi u 2NF jer ima samo jedan ključni atribut i niz atributa.
 - Međutim između atributa brojsobe i telefon postoji međusobna zavisnost (telefon je u sobi) pa će se neki broj telefona ponoviti onoliko puta koliko laboranata radi u datoj sobi. Redundanca!
 - Broj jednog telefona i broj sobe neke laboratorije nije moguće uneti u informacioni sistem sve dok se ne unesu podaci o bar jednom laborantu koji radi u njoj. Anomalija pri upisu podataka!
 - Brisanjem podataka o jedinom laborantu neke laboratorije, gubi se podaci o laboratoriji. Anomalija pri brisanju podataka!
 - Ako laboratorija promeni broj telefona, izmenu treba uneti onoliko puta koliko laboranata radi u njoj. Anomalija pri izmeni podataka.
- **Treća normalna forma (3NF)** Relacija se nalazi u trećoj normalnoj formi ako neključni atributi nisu ni funkcionalno ni tranzitivno zavisni of ključnog atributa.
- Između atributa unutar jedne n-torke može postojati zavisnost, slično kao i kod relacija. Tip te zavisnosti može biti 1:1, 1:N, N:M. Ako je ta

zavisnost između atributa A i B tipa 1:1 onda se ona naziva funkcionalnom zavisnošću i označava se sa $A \rightarrow B$.

- Ako između atributa unutar jedne n-torke postoje sledeće funkcionalne zavisnosti: $A \rightarrow B$, $B \rightarrow C$ i $A \rightarrow C$, onda je atribut C tranzitivno i funkcionalno zavistan od atributa A.
- Da bi se otklonile navedene slabosti kod relacije laborant, relacija mora biti dovedena na treću normalnu formu dekomponovanjem (razlaganjem)
- Dekomponovanje se izvodi tako da se neključni atributi koji su u međusobnoj funkcionalnoj zavisnosti izdvoje u zasebnu relaciju, te da se ta nova relacija poveže sa ostatkom prvobitne preko jednog atributa.
- Konkretno u predhodnom primeru relacije laborant, datu relaciju Laborant <šiflab#, ime, prezime, brojsobe, telefon> treba razložiti na: laboratorija <brojsobe#, telefon>
laborant <šiflab#, ime, prezime, brojsobe>

5.3.2 Problem gubitka informacija

- Međusobne zavisnosti atributa nisu umetnute veze već veze koje postoje u realnom svetu i koje se preslikavaju u informacioni sistem. Tako da ako dekompozicijom ugrozimo te veze informacioni sistem prestaje da bude verodostojan.
- Relacija se dekomponuje bez opasnosti gubitaka funkcionalnih zavisnosti samo ako se dekompozicija vrši prema funkcionalnoj zavisnosti koja ne ide od kandidata ključa.

5.3.3 Ostale normalne forme

- Boyse – Codd-ova normalna forma (BCNF) je u stvari stroži oblik 3NF i izmišljena je da bi ova dvojica još malo pametovali.
- Četvrta normalna forma eliminiše redundancu koja je u 3NF moguća ako među atributima postoji višeznačna zavisnost, čime se eliminišu anomalije pri održavanju.
- Višeznačna zavisnost među atributima neke relacije $R \langle x, y, z \rangle$ postoji onda kada skup vrednosti atributa Y zavisi od X a ne zavisi od Z. Onda X višeznačno određuje Y, a Y višeznačno zavisi od X.
- Peta normalna forma je poslednji stupanj dekompozicije a glasi: Relacija se nalazi u 5NF ako se ne da dalje, sa nekim smislom, dekomponovati.

Glava 7 Osnove projektovanja

- Informacioni sistem se projektuje da bi poboljšao ili izmenio postojeći IS (koji uopšte ne mora biti kompjuterizovan)
- Proces projektovanja IS može se najgrublje podeliti na dva koraka:
 - analiza realnog sistema (preliminarna i detaljna istraživanja)
 - projektovanje baze podataka i izrada aplikacija
- Prvi koraku analizi sistema jeste da se odredi domen, odnosno oblast rada za koju se projektuje IS (računovodstvo, pravosuđe itd.)
- Modifikacija sistema treba da bude rezultat a ne cilj analize tj. ne treba po svaku cenu uvoditi atraktivna rešenja koja ne doprinose efikasnosti sistema

- Za sve sisteme, razlozi uvođenja računara u rad su slični, ako ne i isti:
 - veća brzina rada računara u odnosu na ljude, posebno kada se radi o obradi velikog broja podataka sa mnogo proračuna
 - veća tačnost podataka i njihova obrada uvek na isti način
 - brži pristup podacima i mogućnost dobijanja novih informacija
 - povećanje sigurnosti podataka
 - povezivanje raznih informacionih sistema
 - smanjenje troškova poslovanja
- Projektovanje informacionog sistema (IS) sastoji se od više faza:
 - definisanje izvora informacija
 - izbor metoda i projektovanje sistema
 - izbor modela sistema
 - izbor načina izrade dokumentacije
 - izbor načina održavanja dokumentacije
 - definisanje načina vođenja projekta
 - vrednovanje kvaliteta projekta

7.1 Prikupljanje informacija

- Prikupljanje informacija je jedan od najvažnijih koraka u razvoju informacionog sistema.
- Informacije se prikupljaju od:
 - korisnika odnosno investitora projekta
 - postojeće dokumentacije koja kruži u sistemu
 - spoljnih izvora – literatura i eksperti za određenu oblast
- postupak prikupljanja informacija vrši se prvo u fazi analize postojećeg IS, a potom u fazi uvrđivanja ciljeva koji se žele postići.
- **Izvori informacija.** Osnovni izvor informacija su korisnici sistema.
- Od korisnika se saznaje koji se procesi i postupci izvode u postojećem sistemu, kao i ciljevi koje treba postići novim informacionim sistemom.
- Informacije se od korisnika prikupljaju pomoću intervjua, upitnika i posmatranjem.
- Intervju se koristi za skupljanje podataka u direktnom razgovoru sa korisnikom. Intervju mora biti dobro isplaniran. Analitičar koji planira intervju mora znati koje informacije želi da dobije, kao i koje informacije može i sme tražiti od sagovornika. Takođe izbor sagovornika mora biti takav da garantuje pouzdanos podataka koje dobijamo, tj. sagovornika treba intervjuisati strogo u vezi njegovih nadležnosti. Intervjue treba vršiti kontinualno (u toku određenog vremenskog perioda) i sa raznim osobama. Posebno treba izbegavati računarsku terminologiju koju sagovornik eventualno ne razume a neće to javno da kaže, pa često daje netačne informacije.
- Upitnik (anketa) je tehnika koja omogućava analitičaru da kontaktira veliki broj ljudi i da uporedi njihove odgovore na ista pitanja. Upitnik je najčešće anoniman. Pitanja se postavljaju sa ponuđenim odgovorima, a ako to nije moguće ostavlja se mesto za opširniji odgovor.
- Analizom postojeće dokumentacije, uglavnom «ulaznih i izlaznih» papira koji cirkulišu u toku radnog procesa, može se doći do relevantnih saznanja. Ako je sistem predhodno bio kopjuterizovan u nekoj meri, treba analizirati i postojeći software kao i prateću dokumentaciju (uputstva i sl.) Konačno u

dokumentaciju ulaze svi pisani dokumenti kje se pojavljuju u poslovanju (narudžbenice, računi, magacinske liste, itd.)

- Posmatranje je tehnika koja pomaže da se otkriju činjenice koje se ne mogu utvrditi drugim tehnikama. Obično u jednoj organizaciji dolazi do niza konfliktnih situacija koje se ne mogu videti kroz dokumenta ili o kojima sagovornici nisu spremni da govore, a koje bi mogle biti otklonjene novim informacionim sistemom.
- Spoljnji izvori informacija podrazumevaju spoljnje saradnike, savetnike, eksperte, literaturu kao i slične sisteme koji deluju u okolini. Spoljnji izvori su naročito važni ako se informacioni sistem projektuje po prvi put, pa korisnici sistema nisu sasvim sigurni šta hoće.
- **Izbor metoda** – Uobičajno je da se u okviru jednog projektnog tima koristi jedna, detaljno razrađena metodologija, jer se time postiže standardizacija procesa projektovanja i dokumentovanja sistema.
- Svaki IS određen je sa tri osnovna elementa
 - tokovima podataka u informacionom sistemu
 - procesima u informacionom sistemu
 - podacima u informacionom sistemu
- Svaka metodologija mora da sadrži sredstva koja omogućuju definisanje tokova podataka, procesa obrade i organizacije podataka.
- Jedan od mogućih pristupa rešavanju problema (metoda) projektovanja IS je njegovo modelovanje.

7.2 Izrada modela

- Izrada modela informacionog sistema izvodi se u dva koraka i to:
 - izrada logičkog modela informacionog sistema sa definisanjem logičke strukture sistema, podsistema tokova procesa i modela podatka.
Model procesa i model podataka su konačni rezultati faze istraživanja, koja najčešće započinje analizom stanja postojećeg sistema.
 - izrada fizičkog modela informacionog sistema, projektovanje baze podataka (na osnovu modela podataka) i izrada aplikacija (na osnovu modela procesa).

7.3 Analiza postojećeg sistema

- Projektovanje novog IS uvek započinjemo analizom postojećeg ako takav sistem postoji.
- Analiza postojećeg sistema izvodi se u više faza i to:
 - analiza i prikaz globalne fizičke strukture postojećeg sistema
 - dekompozicija sistema na njegove podsisteme
 - izrada dijagrama toka podataka za svaki podsistem
 - izrada i detaljan opis potrebnih procedura za obradu
- Logički model postojećeg sistema treba da pokaže logičku strukturu pojedinih elemenata ne ulazeći u način izvođenja pojedinih procedura.

7.4 Projektovanje novog sistema

- Projektovanje novog IS izvodi se u tri osnovne faze:
 - Definicija problema
Definisanje ciljeva i spoljnjih ograničenja na sistem

- Izrada okvirnog projekta
Definicija, izrada i oblik logičke strukture novog sistema
Izrada i usvajanje predloga fizičkog modela novog sistema
Specifikacija računarske opreme
- Izrada detaljnog projekta
Definicija korisničkih procedura
Izrada predloga logičkog modela
Definicija baze podataka i programa obrade podataka
- U praksi je ustanovljeno da se prave tri predloga novog sistema, koji se potom daju investitoru na usvajanje:
 - prvi predlog treba da definiše moguću ali ostvarljivu granicu automatizacije novog sistema
 - drugi predlog definiše minimum zahvata u postojeći sistem koji donose kvalitativne novine.
 - Treći predlog treba da bude kompromis želja i mogućnosti
- Svi navedeni predlozi vrednuju se pomoću tri osnovna kriterijuma:
 - tehničkoj izvodljivosti
 - operativno-organizacionoj prihvatljivosti
 - ekonomskoj opravdanosti
- Sistem se smatra tehnički izvodljivim ako u organizaciji postoji oprema, odnosno ako postoji mogućnost nabavke opreme za njegovu realizaciju.
- Operativno-organizacijski prihvatljiv sistem je onaj sistem koji na svom izlazu daje kvalitetne informacije za potrebe tehnološkog procesa i upravljanja radnom organizacijom.
- Ekonomska opravdanost razvoja novog informacionog sistema određuje se poređenjem troškova za njegovu realizaciju i dobiti koju sistem donosi.
- Postupak projektovanja novog sistema polazi od logičkog modela postojećeg sistema, ciljeva koji se žele dostići a definisanih u predhodim fazama i zahteva postavljenih u toku analize postojećeg sistema.
- Izvođenje se realizuje u dve faze:
 - izrada okvirnog projekta
 - izrada detaljnog projekta
- Izrada okvirnog projekta svodi se na definisanje ciljeva i problema, a zatim na ocenu mogućnosti i načina rešavanja.
- Izrada detaljnog projekta sastoji se od definisanja korisničkih procedura i fizičkog modela baze podataka.
- Na osnovu korisničkih procedura i fizičkog modela baze podataka definišu se procesi obrade podataka (pišu se programi).

7.5 Realizacija sistema

- Realizacija sistema sastoji se od:
 - inicijalizacije baze podataka
 - izrade programa i testiranje u realnom okruženju
 - izrade konačne dokumentacije
 - uvođenja sistema u eksploataciju
- Inicijalno punjenje baze podacima je prva faza realizacije. Tek nakon punjenja baze određenom količinom podataka možemo pratiti da li se programi ponašaju kako treba.

- Izrada i testiranje programskih komponenti sistema izvodi se tako što se svaki završeni modul odmah testira kako bi se izbeglo ponavljanje grešaka u sledećim modulima.
- Na kraju sledi testiranje informacionog sistema kao celine. U realnoj situaciji proveravaju se programi i model procesa, a najčešće se rezultati upoređuju sa rezultatima starog IS-a.
- Problemi koji se mogu javiti su posledica grešaka u izradi modela procesa, modela podataka i grešaka u njihovoj implementaciji.
- Izrada i održavanje dokumentacije sistema deli se na
 - izvještaje po fazama rada
 - kompletnu dokumentaciju
- Izveštaji služe za informisanje investitora i treba da sadrže:
 - specifične podatke o radu i rezultatima za svaku fazu rada
 - opšta saznanja o projektu stečena u svakoj fazi rada
 - preporuke i prijedlog nastavka rada
 - izvještaj o utrošenim sredstvima
 - dokumentovan zahtev za sredstva potrebna za nastavak
- Kompletna dokumentacija sistema treba da sadrži:
 - opis strukture sistema
 - opis procesa sistema
 - opis strukture podataka
 - uvod i osnovne karakteristike sistema
 - ulaz i izlaz iz sistema
 - osnovne funkcije sistema
 - detaljan opis instalacije sistema
 - uputstvo za upotrebu

7.6 Vođenje i vrednovanje projekta

- Uspešnost vođenja projekta zavisi od više faktora a najvažniji su:
- Sastavljanje projektnog tima je često presudan faktor realizacije IS
- Projektni tim treba da se sastoji od: analitičara, tehnologa, projektanta, korisnika, rukovodioca radne organizacije i predstavnika investitora.
- Vrednovanje projekta izvodi se na osnovu niza kriterijuma od kojih su najvažniji:
 - ispravnost, da li sistem daje tačne informacije
 - potpunost, da li sistem može da generiše sve potrebne informacije
 - robusnost, kako sistem reaguje na nepredvidivo rukovanje
 - pouzdanost, je pre svega određena kvalitetom angažovane opreme
 - optimalnost, da li je sistem mogao biti realizovan sa manje para
 - jednostavnost u korišćenju, koliko je sistem user friendly
 - jednostavnost održavanja, jednostavnost održavanja i izmene sys
 - mogućnost proširenja, ili povezivanja sa drugim sistemima
 - prenosivost, fleksibilnost sistema i prenošenje u druge uslove
- Iskustvo je pokazalo da IS nikada u potpunosti ne zadovoljavaju postavljene zahteve. Razlog tome nije, kako se obično misli, nedovoljno dobro izvršena analiza sistema ili loše definisani zahtevi. Pravi razlog leži u tome da se podaci i procesi u realnom sistemu stalno menjaju i zato informacioni sistemi moraju biti sposobni da se prilagođavaju i nadograđuju.