

# Osnove baza podataka pitanja:

## 1. Istorijski razvoj automatizovane obrade podataka:

Nastankom ENIAC-a 1950ih nastaje termin automatizovana obrada podataka. Racunari su se u pocetku uglavnom koristili za resavanje nekih naucnih i racunskih problema nad relativnom malim skupom podataka, ali vremenom javlja se potreba za obradom velike kolicine podataka, kao sto je bila u finansijama npr. , i ubrzo krece koriscenje memorijskih jedinica i novih metoda pristupa za cuvanje ali i obradu podataka.

## 2. Klasicna obrada podataka(prednosti,mane,nastajanje modela podataka):

Klasicna obrada podataka podrazumeva cuvanje svih podataka u neakvim fajlovima(txt npr).

Njena prednost jeste jednostavna realizacija i projektovanje za jednu aplikaciju, ali mane jesu to sto je vrlo komplikovano projektovati takav sistem za vise aplikacija koje bi koristile isti izvor podataka(nepovezanost podataka),

to sto bi mi ako imamo vise aplikacija koje koriste iste podatke imali iste fajlove za svaku aplikaciju sto bi dovelo do postojanja vise datoteka sa istim podacima(redundantnost podataka),

takodje mana je i cvrsta povezanost programa sa datotekom jer ako izmenimo fizicku ili logicku strukturu te datoteke moramo menjati logiku svakog programa koji je koristi.

Takodje vremenom kod slucaja da postoji vise programa moze se desiti da se dodje do neusaglasanosti izmedju datoteka gde za entitete sa istim obeležjima mozemo imati razlicite podatke.

Model podataka i struktura datoteka nastaje iz zahteva tj iz nasih potreba nezavisno od samog programa na osnovu objekata i njihovih veza koje treba da podrzimo.

## 3. Konceptcija baza podataka(kada? Koji je cilj?):

Zbog samih problema koje je sadrzala klasicna obrada podataka krajem 50ih se odlucilo da bi sve podatke trebalo cuvati u nekakvoj bazi podataka. Sam koncept baze podataka ima za cilj da :

- Sve podatke jednog informacionog sistema treba integrisati u jednu fizicku strukturu podataka

- Svi aplikacioni programi pri obradi baze podataka koriste standardne softverske alate nazvane sistemima za upravljanje bazom podataka(SUBP)

#### **4. SUBP? (Sistem za upravljanje bazom podataka):**

SUBP je program namenjen rukovodjenju bazom podataka odnosno skupom strukturiranih podataka kao i da izvršava operacije nad njima. Obezbeđuju pretraživanje, čuvanje, ažuriranje, sigurnost podataka itd.. Osnovni zahtev jeste očuvanje integriteta podataka tj čuvanje tačnosti i koinzistentnosti istih. Neke od poznatijih su Oracle, MS SQL Server, Postgres, MySQL

#### **5. Poslovni zadaci Administratora baze podataka:**

Poslove kao što su regulisanje prava pristupa, pravljenje sigurnosnih kopija, koinzistentno sprovođenje promena itd. Ne možemo poveriti običnom korisniku. Zbog toga se javlja pojam Administratora baze podataka. Administrator baze podataka je osoba odgovorna za specifikiranje, projektovanje, implementiranje, efikasan rad i održavanje baze podataka. Administrator baze podataka radi na sledećim poslovima:

- Radi sa korisnicima pri uspostavljanju zahteva za bazu podataka
- Definise bazu podataka
- Radi sa programerima koji treba da pristupaju bazi podataka
- Odgovoran je za punjenje baze podacima
- Nadgleda rad baze podataka kako bi ustanovio da li je potrebna neka izmena ili drugacija organizacija podataka

#### **6. Integritet baze podataka:**

Integritet baze podataka je stanje baze podataka u kome su sve vrednosti podataka korektne tako da oslikavaju stanje realnog sveta u zadatoj preciznosti i tačnosti kao i da svi podaci postuju uzajamna pravila konzistentnosti tj. međusobne odnose. Za održavanje integriteta BP zadužen je administrator uz pomoć SUBP.

#### **7. Bezbednost BP:**

Bezbednost baze podataka podrazumeva zaštitu i specavanje od neovlašćenog pristupa podacima kao i nedozvoljenog menjanja istih. Ona je potrebna zbog zaštite od nenamernih ali i namernih pristupa

poverljivim podacima. Administratori moraju voditi računa o pravima pristupa jer može doći do konflikta pri pristupanju bazi imajući na umu da više različitih aplikacija može da ima zajedničku bazu podataka.

## **8. Principi rada sistema za upravljanje bazom podataka:**

Softver za upravljanje bazom podataka je veoma složen i sadrži veliki broj programa i rutina. Posедуje komponente i mehanizme za

- upravljanje transakcijama
- upravljanje višekorisničkim režimom rada
- zaštitu BP od neovlašćenog pristupa
- zaštitu BP od uništenja / oštećenja
- implementaciju šeme BP u izabranom modelu podataka
- formiranje, korišćenje i ažuriranje BP
- upravljanje podacima

On se sadrži od :

- Fizickog nivoa – definiše način na koji su podaci organizovani na spoljnim memorijama
- Konceptualnog nivoa (sema baze podataka)- definiše logičku strukturu baze podataka, sve podatke u sistemu, njihove odnose i veze.. Treba da omogući upravljanje podacima kao zajedničkim resursima u celom sistemu.
- Korisničkog nivoa – definiše logičku strukturu podataka pogodnu za programe.

## **9. Modeli podataka(Redudansa?):**

Model podataka je skup međusobno povezanih podataka koji opisuju objekte, njihove veze i osobine realnog sistema. U modelu podataka ne opisuje se potpuni skup znanja o sistemu, već se vrši odabir i opis relevantnih karakteristika sistema.

Redudansa predstavlja ponavljanje istih podataka u različitim datotekama.

## 10. Apstrakcije podataka:

Apstrakcija predstavlja kontrolisano dodavanje detalja u sistem tj. Skrivanje nepotrebnih detalja i uključivanje samo onih neophodnih. Pri izradi modela podataka koristimo sledeće tipove apstrakcija:

- **Klasifikacija objekta i uzrokovanje**- Objekti koji imaju isti skup osobina (atributa i veza) i isto ponašanje mogu se predstaviti tipom ili klasom. Primer: STUDENT za svakog studenta na fakultetu. Uzrokovanje predstavlja prikazivanje jednog pojavljivanja(uzroka) datog tipa ili klase.
- **Generalizacija i specijalizacija objekata** - apstrakcija u kojoj se skup sličnih tipova objekata predstavlja opštijim generičkim tipom odnosno nadtipom. Slični su oni tipovi objekata koji imaju neke zajedničke osobine i ponašanje. Primer: STUDENT, PROFESOR, CISTACICA mogu se predstaviti kao GRADJANIN npr. Obrnut postupak generalizacije je specijalizacija.
- **Agregacija i dekompozicija** - apstrakcija kojoj se skup objekata i njihovih međusobnih veza tretira kao novi, jedinstveni, agregirani tip koji je na visem nivou apstrakcije. Inverzan postupak agregaciji jeste dekompozicija.

## 11. Komponente modela podataka:

Model podataka sadrži tri komponente:

- **Strukturna komponenta** – čini je skup primitivnih koncepata i skup pravila za izgradnju složenih koncepata.
- **Integritetnu komponentu** – čini je skup uslova integriteta tj ograničenja. Uslovi integriteta ograničavaju pojave baze podataka na samo one koje zadovoljavaju neko od ograničenja kao što je dozvoljena vrednost podataka, dozvoljenih veza između objekata i sl.
- **Operacijsku komponentu** – čini je skup koncepata koji omogućavaju interpretaciju dimenzionalnih karakteristika skupa podataka i promenu stanja podataka u bazi podataka u skladu sa promenom stanja u realnom sistemu. Primer može biti upis godine jednog studenta koji želimo da se izmeni kao što se izmenilo i u realnom sistemu.

## 12. Generacije modela podataka:

Istorijski gledano postoje 3 generacije modela podataka:

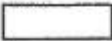
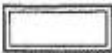


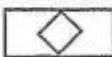


- **Modeli podataka prve generacije** – Svaki konvencioni programski jezik koji ima zaseban model podataka. U nekim jezicima je moguće delimično primeniti apstrakciju generalizacije kao i dodati ograničenja nad vrednostima određenih tipova eksplicitno. Samim tim modeli podataka prve generacije i njihovi programski jezici nisu dovoljno pogodni za modelovanje realnog sistema, te im je praktična primena jako ograničena.
- **Modeli podataka druge generacije** – Za prezentaciju podataka sadrže koncepte kao što su: stablo, set, relacija i dr. U osnovi koriste iste apstrakcije kao modeli prve generacije pa ne postoji potpuna podrška za koncepte agregacije i generalizacije. Ovde je moguće eksplicitno definisati ograničenja na vrsti podatka. Uprkos tome što jesu semantički bogatiji od modela prve generacije i dalje ne mogu dati zadovoljavajući prikaz realnog sistema. Ovoj generaciji pripadaju: funkcionalni model podataka, hierarhijski, mrežni, relacioni ...
- **Modeli podataka treće generacije** – Sadrže koncepte generalizacije i agregacije. Podržavaju sve vrste apstrakcija, poseduju mogućnost eksplicitnog definisanja ograničenja na vrednosti podataka. Dobro modeliraju realni sistem. Korisniku su razumljivi. Ovoj generaciji pripadaju: model objekti-veze (MOV), semantički model podataka, prošireni relacioni model...

## 13. Model objekti veze (ER-model):

Najpopularniji i u praksi najčešće korišćeni. Koristi se kao grafički jezik za projektovanje konceptualne šeme baze podataka.

- **Konceptualna šema:**  
Konceptualna šema predstavlja takav model realnog sistema i njegove baze podataka koji ne zavisi od konkretnog SUBP. Može se lako prevesti u šemu baze podataka na kojoj je SUBP zasnovan. Konceptualna šema MOV predstavlja se uz pomoć dijagrama Objekti-Veze (DOV):

- Tip objekta se predstavlja pravougaonikom sa upisanim nazivom.
- Atributi elipsom sa upisanim nazivom povezani sa odgovarajućim tipom objekta ili tipom veze.
- Atributi kojima se identifikuje objekat se podvlače.
- Broj tipova objekata koji učestvuje u vezi definiše red veze
- Veza se predstavlja rombom.

GRAFIČKI PRIKAZ (simbol)	OPIS
	osnovni objekt
	slab objekt
	veza
	veza nadtipa i podtipa
	mešoviti tip objekta
	obeležje
	linija za povezivanje

Slika 7.1. Grafički simboli za crtanje DOV

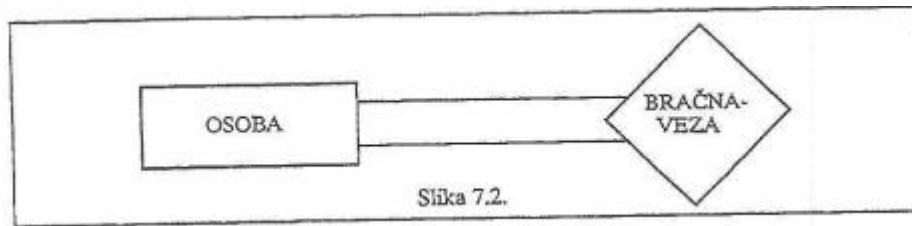
- **Strukturna komponenta:**

Strukturnom komponentom definiše se skup primitivnih koncepata i skup pravila za izgradnju slozenih koncepata. Primitivni koncepti u MOV su objekat, veza(izmedju objekata) i obeležje objekta. Koristeći apstrakciju klasifikacije pojedini objekti se mogu klasificirati u tipove objekata. Objekti se koji se identifikuju nezavisno od ostalih objekata u sistemu se nazivaju osnovni tj jaki objekti i oni se predstavljaju pravougaonikom sa nazivom unutar njega.

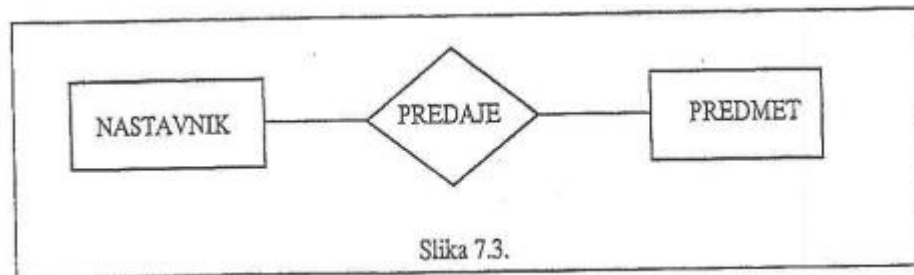
- **Koncept veze:**

Veza u modelu opsije način povezivanja objekata. Broj tipova koji učestvuju u vezi definiše red veze. Možemo razlikovati rekurzivne, binarne i veze viseg reda.

**Rekurzivna veza** - je veza izmedju dva objekta istog tipa.



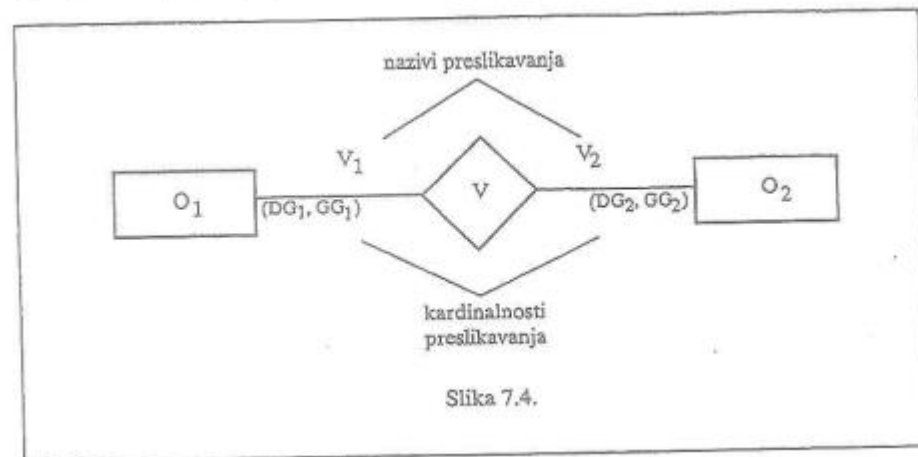
**Binarna veza** – cine je povezani objekti razlicitog tipa.



**Veze viseg reda** - mogu se predstaviti pomocu binarnih veza ili se takva veza tretira kao agregacija.

- **Kardinalnost:**

Informacije o prirodi odnosa izmedju objekata iz realnog sveta u modelu podataka iskazuje se kardinalitetom tipa veze.



Veza V izmedju pojava tipova objekata  $O_1$  i  $O_2$  definiše dva tipa preslikavanja:

$V_1: O_1 \longrightarrow P(O_2)$  preslikavanje sa skupa pojavljivanja  $O_1$  u skup pojavljivanja  $O_2$  ( $O_1$  je domen, a  $O_2$  kodomen preslikavanja),

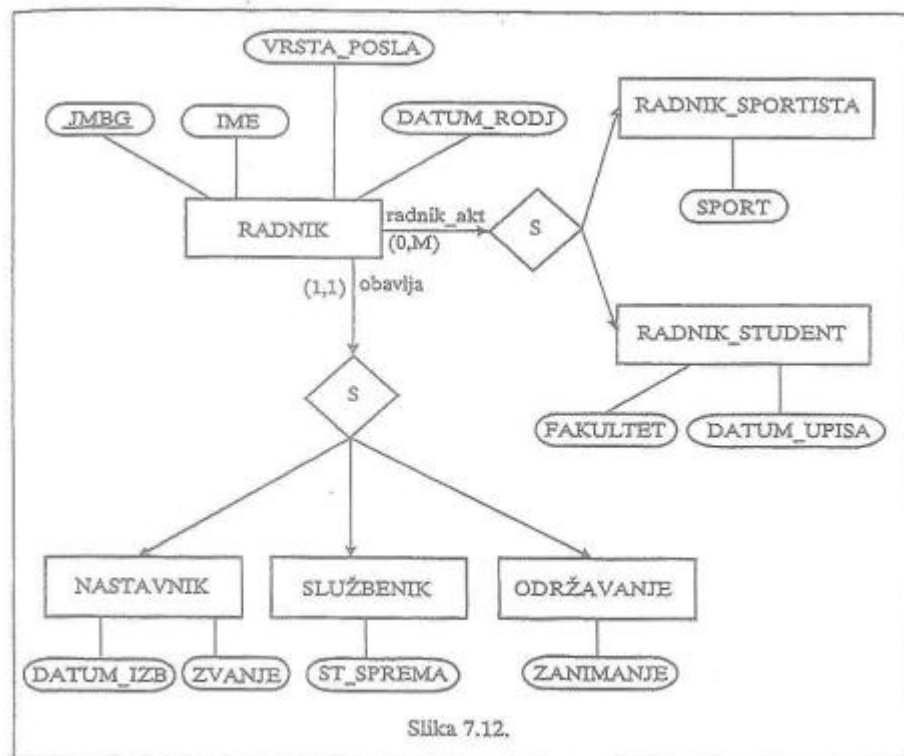
$V_2: O_2 \longrightarrow P(O_1)$  (inverzno) preslikavanje sa skupa pojavljivanja  $O_2$  u skup pojavljivanja  $O_1$  ( $O_2$  je domen, a  $O_1$  kodomen preslikavanja),

gde su:  $V_1$  i  $V_2$  - nazivi preslikavanja,  
 $P(O)$  - partitivni skup skupa  $O$ .

Za svaki od preslikavanja definiše se par (DG,GG) gde DG(Donja granica) predstavlja najmanji a GG(Gornja granica) najveći broj elementa skupa u koji se preslikava jedan element skupa originala. Par (DG,GG) predstavlja kardinalnost preslikavanja. Ako donja granica ima vrednost nula onda je to parcijalna(tj opcionalna)veza, dok ako je taj broj veći od nule onda je ta veza totalna(obavezna). Gornja granica može imati bilo koju vrednost veću od jedan koja nije poznata i to se označava sa M. Isto tako mora se ispostovati da  $DG \leq GG$ .

- **Nadtip – podtip:**

Kao što već znamo generalizacija je apstrakcija u kojoj se skup sličnih tipova objekata tretira kao jedan generički tip objekata tj. Nadtip. Specijalizacija predstavlja inverzan postupak u kome se za neki tip objekta definišu njegovi podtipovi koji imaju neka samo njima specifična obeležja, veze ili operacije. Podtipovima se onda pripisuju sva obeležja koja ima i njihov nadtip a može se dodati i ona njima specifična. Možemo reći da je neki objekat podtip drugog objekta ako je svaka njegova pojava ujedno i pojava tog drugog objekta.

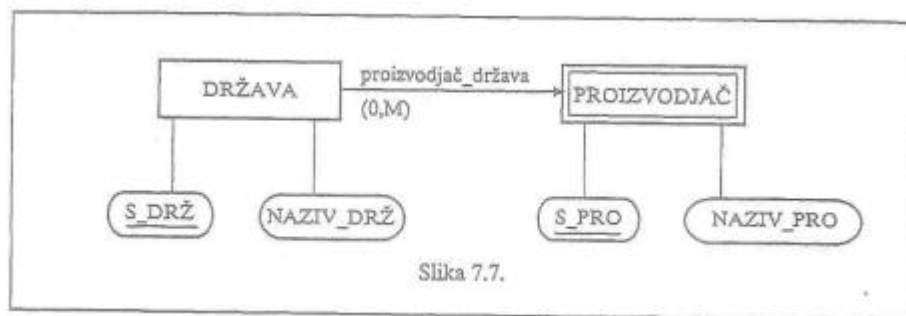




Takodje s obzirom na DG i GG mogu se definisati dve vrste specijalizacije:  
 onda kada svako pojavljivanje tipa spezijalziuje se u tacno jedan podtip(DG=GG=1)  
 i kada jedno pojavljivanje nadtipa moze uzrokovati vise pojavljivanja podtipa(DG=0 , GG>1 tj. GG= M).

- **Slabi objekat**

Tip slabog objekta je takav tip objekta koji nema vlastitih obelezja za indetifikaciju, vec koristi obelezja nekog drugog tipa objekta koji moze biti jak(osnovni) ili slab tip. Kaze se da tip slabog objekta indefitikaciono zavisi od nekog drugog tipa objekta

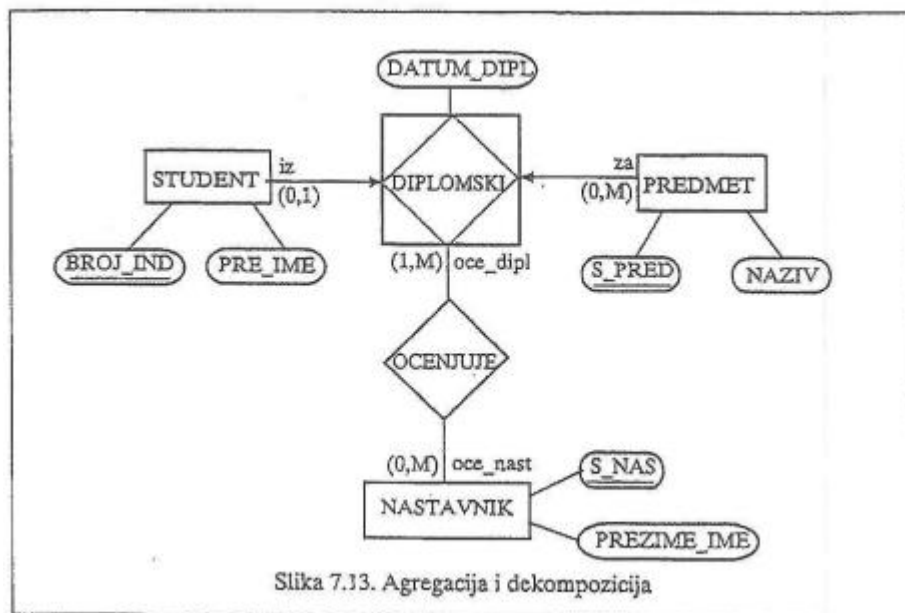


Oznacavaju se upisivanjem njegovog naziva u dvostruki pravougaonik, i samim tim da slab obejakt ne moze postojati bez njegovog nadredjenog objekta podrazumeva se da ce donja granica preslikavanja biti 1. To znaci da jedna pojva tipa slabog objekta u vezi je samo sa jednim tipom nadredjenog objekta. Ako se u bazi podataka brise pojava nekog objekta tada se brisu i sva pojavljivanja njemu indetifikacioni zavisnih slabih objekata.

- **Agregacija:**

Na MOV vezu je moguće uspostaviti samo između dva tipa objekta. Kada bi nam bilo potrebno da tip veze ima neka obeležja ili da se uspostavi veza između tipa veze i tipa objekta tada tip veze postaje mesoviti tip objekt-veza. Mesoviti tip objekta može imati i obeležja i povezati se sa drugim tipovima objekta. Prevođenje tipa veze u tip objekta izvodimo apstrakcijom agregacije u kojoj se veza između dva ili više tipova objekta tretira kao objekat na visem nivou apstrakcije. Agregatni tip objekta se razlikuje od ostalih objekata po tome što nema svoj indetifikator već ga indetifikuju objekti koje on agregira. Suprotan proces

agregaciji bio bi dekompozicija i njeno preslikavanje je uvek trivijalno(1,1) pa se na modelu i ne navodi.



#### 14. Proces izrade MOV:

Izrada MOV za dati realni sistem se obicno odvija u sledecim koracima:

- Izrada MOV po delovima
- Integracija delova u jednu celinu koja se naziva globalnim modelom podataka
- Uklucivanje drugih semantickih detalja u model kao sto su uslovi integriteta

Proces izrade MOV je iterativan tj. Svaka naknadna saznanja i zahtevi cesto mogu dovesti do izmene postojećeg rešenja. Takođe prilikom izrade postoje razni obrasci za koje se preporučuje da ih koristimo.

##### **Integriranje podmodela:**

Kada smo već završili izradu po delovima tj. Podmodelima pristupa se integrisanju tih podmodela u jedan globalni model podataka. U tom procesu često može doći do novih saznanja pa se vrlo često javlja potreba za izmenom u pojedinim podmodelima. Pre same integracije treba proveriti da li se u pojedinacnim modelima javljaju identični objekti ili neki sinonimi ili homonimi. Sve iste ili slične tipove treba predstaviti jednim globalnim modelom tipa kao što i sve sinonime treba zameniti sa jednom rečju a homonime promeniti tako da svaki tip ima novi jedinstven naziv. Tek nakon ove faze možemo preći na pravu implementaciju i korišćenje svih koncepata apstrakcije.

**Golobalni model podataka treba da bude:**

Globalni model podataka mora biti:

- kompletan(sadrzi sva znanja iz svih podmodela)
- neredundatan(ne sadrzi visestruke predstave istih znacjenja)
- koinzistentan(ne sadrzi modjusobno protivrecna znacjenja tj da na razlicitim podemodelima imamo medjusobno protivrecne tvrdnje o nasem sistemu)

**15. Relacioni model podataka:**

Koncept realcionog modela podataka je prvi put objavljenja u clanku Edgara Codd-a iz 1970 godine.Nastao je sa ciljem da se rese problemi koji su imali tada poznati modeli podataka.Zasnva se na relacionoj algebri.Osnovni pojam realcionog modela je realcija.Relacija se moze posmatrati iz dva aspekta znacenje i sadrzaj.Znacenje realcije se naziva intezijom i formalno se iskazuje semom relazije.Sadrzaj relacije naziva se ekstenzija i iskazuje se tabelom podataka ili pojavom seme relacije.

**Relaciona sema(Sema relacije):**

Sema realcije tj. Zaglavlje tabele nam daje apstraktnu ideju o tome sta moze biti u nasoj tabeli.Sema realcije definise samu strukturu baze.Ona se definise nazivom,skupom obelezja i ogranicenjima.

**Relacija:**

Prema matematickom shvatanju relacija nad neka dva skupa je podskup njihovog dekartovog proizvoda.Relacija zapravo predstavlja citavu strukturu(tabelu) u kojoj se podaci predstavljaju i organizovana je u redove i kolone. Svaki red naziva se n-torka a svaka kolona n-torke atribut.Ona nije uredjena tako da redosled n-torki nije bitan.

**Koje uslove treba da ispuni skup obelezja da bi bio koandidat za kljuc seme relacije:**

S obzirom da je svaka n-torka relacije jedinstvena to znaci da ce postojati jedinstveni indetifikator istih.Taj indetifikator se naziva kandidatom kljuca.Da bi jedan skup obelezja bio kandidat mora ispuniti sledece uslova:

Jedinstvenost - Ni u jednoj ekstenziji relacije ne postoje n-torke koje imaju jednake vrednosti svih njegovih obelezja.

Minimalnost – broj obelezja u kljucu bi trebalo svesti na mimumum.

Jedan od kandidata za kljuc se bira za primarni, a oni koji ispune uslov jedinstvenosti ali ne i uslov minimalnosti se nazivaju super kljuc(npr. jmbg i ime).

**Spoljni kljuc:**

Spoljni kljuc su ona obelezja ili skup obelezja koji nisu primarni kljucevi date tabele vec se nalaze u nekoj drugoj semi relacije.(Neke medjutable npr)

**16. Integritetna komponenta relacionog modela podataka:**

**Problem nedostajucih informacija:**

Model baze podataka polazi od pretpostavke da su sve relevante informacije poznate i da se mogu uneti u bazu podataka.U praksi ova pretpostavka cseto nije ispunjena.Taj problem se naziva problem nedostajucih informacija.Ovaj problem se pojavljuje kada npr. u trenutku unosa podataka u bazu nisu poznati neki podaci(adresa boravka,broj telefona i sl.)Nekada ce biti moguće te informacije dobiti naknadno ali nekada to nije moguće kao npr naci ime supruga osobi koja nije u braku.

**Null vrednosti:**

Na mesto koje treba upisati stvarnu vrednost obelezja koje nam nije poznato mi upisujemo null vrednost.Null nam oznacava da ta vrednost nije poznata.To moze komplikovati operacijsku komponentu pa se gleda da se izbegnu kada god je to moguće.

**Integritet baze podataka(kako se obezbedjuje,podela):**

Pod integritetom baze podataka podrazumeva se istinitost i ispravnost podataka u njoj.On se definise uslovima ili pravilima integrita nad sadrzajem baze podataka organicavajući je na neka dozvoljena stanja.Uslowe integriteta mozemo podeliti u dve grupe:

Opsti uslovi integriteta – vase u svim relacionim bazama podataka.

Korisnicka pravila integriteta – specificna za pojedine aplikativne sisteme,tj u oblasti primene.

**Od cega zavise korisnicki uslovi integriteta:**

Oni zavise od korisnika,specifikacije.mogucnosti...

**Opsti uslovi integriteta koji vase u svim realcionim bazama:**

Pravila integriteta mogu biti staticka i dinamicka.Statickim pravilima definisemo uslove koji moraju vaziti pre i posle izvorsavanja bilo koje operacije nad bazom, dok dinamicnim pravilima definisane su procedure u relacionom modelu koje ce nam garantovati ostvarenje uslova

integriteta. Relacioni model podataka postuje sledeca staticka pravila integriteta:

- Integritet objekta
- Referencijalni integritet.

#### **Sta je i kako glasi integritet objekta:**

Integritet objekta vezan je za sam pojam primarnog ključa same relacije. S obzirom na osobine primarnog ključa sledi definicija da vrednost primarnog ključa kao celine kao i svaka njegova komponenta ne sme biti jednaka null vrednosti. To se ne sme desiti jer nam primarni ključ omogućava lako i efikasno adresiranje n-torke pa ako se tu nađe null vrednost ne bi smo mogli to ostvariti.

#### **Sta je i kako glasi referencijalni uslov integriteta:**

Referencijalni integritet predstavlja poseban slučaj opštijeg uslova integriteta, koji se naziva zavisnost sadržavanja. Zavisnost sadržavanja definiše ograničenje u smislu da se u nekoj relaciji ne može upisati pozivajuća n-torka ako u drugoj relaciji ne postoji bar jedna ciljane n-torka. Pri brisanju ciljane n-torke moraju se brisati i sve pozivajuće n-torke. Na primer ako imamo tabelu nastavnika koja bi pozivala katedru na kojoj predaje preko sifre katedre ako bi obrisali katedru onda nam sifra kat iz nastavnika ne pokazuje ništa i narušava referencijalni integritet. Taj problem nam rešava SUBP koji će nam prijaviti grešku u tom slučaju.

### **17. Operacijska komponenta relacionog modela:**

#### **Na čemu se zasniva:**

Operacijska komponenta omogućava manipulisanje podacima u bazi podataka što se pre svega odnosi na postavljanje upita i azuriranje baze podataka. Smisao operacija nad semom baze podataka jeste u izmeni postojeće same relacije kao i u dodavanju novih relacija. Svi postojeći jezici relacione baze podataka kao što je SQL su razvijeni na osnovu:

- relacione algebre
- relacionog računa

#### **Definicija operacija relacione algebre:**

Operacije relacione algebre mogu se podeliti u dve grupe.

Prvu grupu čine operacije matematičke teorije skupova:

- UNIJA,
- RAZLIKA,

-PRESEK,

-DEKARTOV PROIZVOD,

i u relacionom modelu podataka se mogu primeniti pošto je relacija definisana kao skup n-torki.

Drugu grupu operacija čine operacije koje su posebno razvijene za relacioni model podataka:

-SELEKCIJA,

-PROJEKCIJA

-SPOJ.

Rezultat svake operacije nad nekom relacijom ili skupom relacija je nova relacija nad kojom se dalje mogu primenjivati operacije relacione algebre.

### Binarne operacije(UNIJA, PRESEK I RAZLIKA):

Mogu se izvoditi samo na relacijama koje su međusobno uporedive tj kompatibilne. Relacije su kompatibilne ako imaju isti broj oblezja i imaju domen iste vrste. Medju tim operacijama spadaju:

#### **Operacija unije:**

Unija izmedju dve relacije je skup n-torki koji su sadržani u jednoj od relacija ili i obe relacije  $r \cup p = \{t \mid t \in r \vee t \in p\}$ .

		$r \cup p(A \ B \ C)$		
		a	b	c
		d	e	f
		c	b	f
		a	b	f
b)	$r(A \ B \ C)$	a	b	c
		d	e	f
		c	b	f
	$p(C \ A \ B)$	c	a	b
		f	a	b

Ona ima osobine komutativnosti i asocijativnosti( $A \cup B = B \cup A$ ,  $(A \cup B) \cup C = A \cup (B \cup C)$ )

#### **Operacija preseka:**

Presek dve kompatibilne relacije je relacija koja sadrži sve n-torke koje su istovremeno elementi i prve i druge relacije.  $r \cap p = \{t \mid t \in r \wedge t \in p\}$ .

$r(A\ B\ C)$	$p(A\ B\ C)$	$r \cap p = r' (A\ B\ C)$
a b c	a b c	a b c
d e f	a b f	e f g
e f g	e f g	
a b g		

### Operacija razlike:

Razlika izmedju dve kompatibilne relacije je skup svih n-torki sadrzanih u prvoj koji nisu sadrzani u onoj drugoj.  $r - p = \{t \mid t \in r \wedge t \notin p\}$ .

$r(A\ B\ C)$	$p(A\ B\ C)$
a b c	a b c
d e f	a b f
c b f	
$(r - p)(A\ B\ C)$	$(p - r)(A\ B\ C)$
d e f	a b f
c b f	

### Operacija dekartov proizvod:

Dekartov proizvod dve relacije predstavlja skup n-torki koje su nastale kao rezultat spajanja svake pojedine n-torke sadrzane u relaciji jedan sa svakom pojedinom n-torkom iz relacije dva.  $r \times p = \{(t_r, t_p) \mid t_r \in r \wedge t_p \in p\}$ .

Relaciona sema dobijena dekartovim proizvodom za obelezja ima uniju skupa obelezja obe relacione seme. Ako je presek skupova obelezja prazan skup tada dekartov proizvod jeste relacija a u suprotno to ne bi bilo tacno jer bi se pojavljivala ista obelezja sto nije dozvoljeno. Za razliku od dekartovog proizvoda nad skupovima dekartov proizvod nad relacijama je komutativan jer se redosled n-torki moze menjati.

a) Neka vredi  $R \cap P = \emptyset$ . Dekartov proizvod  $r \times p$  je relacija.

$r(A \ B)$	$p(C \ D)$	$(r \times p)(A \ B \ C \ D)$
a b	b c	a b b c
c d	d e	a b d e
a c		c d b c
		c d d e
		a c b c
		a c d e

b) Neka vredi  $R \cap P \neq \emptyset$ . Dekartov proizvod  $r \times p$  nije relacija.

$r(A \ B)$	$p(B \ D)$	$(r \times p)(A \ B \ B \ D)$
a b	b c	a b b c
c d	d e	a b d e
a c		c d b c
		c d d e
		a c b c
		a c d e

### Operacija selekcija:

Selekcija je unarna operacija kojom se iz relacije izdvaja odredjeni podskup n-torki koji ce zadovoljavati odredjeni uslov koji smo mi definisali.

$$\sigma_F(r) = \{t \mid t \in r \wedge t \text{ zadovoljava } F\}.$$

Kriterijum za selekciju moze sadrzati: konstante, nazive obelezja relacije nad kojom se selekcija izvodi, operatore poredjenja ( $=$ ,  $\neq$ ,  $<$ ,  $\leq$ ,  $>$ ,  $\geq$ ), logicke operatore i slcino.

$r(A \ B \ C)$	$\sigma_{A=3}(r)(A \ B \ C)$	$\sigma_{B=C}(r)(A \ B \ C)$	$\sigma_{B < C}(r)(A \ B \ C)$
a 1 2	a 1 2	a 5 5	a 1 2
d 3 4	a 5 5		d 3 4
a 5 5			

### Operacija projekcija:

Projekcija predstavlja unarnu operaciju u kojoj se iz date relacije izdvajaju pojedine kolone:

$$\pi_X(r) = \{t[X] \mid t \in r\}.$$

$r(A \ B \ C)$	$(A \ B)$	$\pi_{AB}(r) = r'(A \ B)$
a b c	a b	a b
d e f	d e	d e
a b f	a b	



### Operacija spoj:

Operacija spoj je složena binarna operacija za koju se može reći da se izvršava u tri koraka:

1. Formiranje dekartovog proizvoda relacija.
2. Iz dekartovog proizvoda se izdvajaju n-torke koje zadovoljavaju postavljene uslove
3. Iz tabele dobijene u drugom koraku izdvajaju se određene kolone(samo u slučaju prirodnog spoja)

$r(A\ B)$	$p(C\ D)$
a 2	4 d
e 5	1 f
f 2	
1. Korak: $(r \times p) = r' (A\ B\ C\ D)$	
a 2 4 d	
a 2 1 f	
e 5 4 d	
e 5 1 f	
f 2 4 d	
f 2 1 f	
2. Korak: $\sigma_{B < C}(r') = r'' (A\ B\ C\ D)$	
a 2 4 d	
f 2 4 d	

Theta spoj( $\theta$  - spoj) predstavlja najopstiji oblik operacije spoja

Teta spoj  $r[A_i \theta B_j]$  je skup n-torki koji zadovoljava sledeće uslove:

- podskup je dekartovog proizvoda tih relacije
- svaki element tog podskupa zadovoljava uslov izdvajanja

$$r[A_i \theta B_j]p = \{(t_r, t_p) \mid t_r \in r \wedge t_p \in p \wedge t_r[A_i] \theta t_p[B_j]\}.$$

Theta spoj će biti relacija ukoliko je i njegov dekartov spoj relacija. U tom slučaju spoj se može definisati na osnovu dekartovog proizvoda i

selekcije  $r[A_i \theta B_j]p = \sigma_{A_i \theta B_j}(r \times p).$

Postoje dva slučaja teta spoja: kada uslov izdvajanja ne postoji(tada operacija prelazi u dekartov proizvod), kada je operator znak jednakosti pa se to naziva spoj sa izjednačavanjem(equi-join).

$r(A\ B)$	$p(C\ D)$	$r[B = C]p = q(A\ B\ C\ D)$
a 6	1 d	a 6 6 f
b 1	6 f	b 1 1 d
e 3		e 1 1 d
e 1		

### Operacija prirodnog spoja:

Prirodni spoj je u tesnoj vezi sa teta spojem. Kao rezultat dobije se tabela sa najmanje dve kolone indetichnih vrednosti obelezja, a cesto su i nazivi identicni. Da bi smo tu tabelu transofrmisali u realciju moramo odstraniti kopije obelezja zajdeno sa odgovarajucim zaglavljima tabele. Ova dodatna operacija se izvodi u okviru operacije prirodni spoj.

Prirodnim spojem dve relacije spajaju se međusobno n-torke tih relacija na osnovu vrednosti obeležja koja se nalaze u obe šeme relacija.

Operaciju prirodnog spoja mozemo izraziti pomocu jednostavnih operacija: Dekartovog prozivoda, selekcije i projekcije.

*Primer:* Neka su  $r(R)$  i  $p(P)$  relacije i neka vredi  $R = \{A, B, C\}$ ,  
 $P = \{B, C, D\}$  i  $T = \{A, B, C, D\}$ .

$r(A\ B\ C)$	$p(B\ C\ D)$	$r \nabla p = q(A\ B\ C\ D)$
a 2 4	2 7 d	a 2 4 k
b 1 5	1 5 h	b 1 5 h
c 2 7	2 4 k	c 2 7 d

Mogu postojati dva slucaja prirodnog spoja:

-Kada je presek obelezja prazan skup i to znaci da one nemaju zajednickih obelezja pa je prirodni spoj tada jedank dekartovom proizvodu

Kada su skupovi obelezja jednaki(obe relacije su zadate na istoj realcionoj semi) tada je rezultat prirodnog spoja jednak preseku relacija.

### SQL(osobine):

Sql nije samo upitni jezik vec predstavlja kompletan jezik podataka koji sadrzi jezike za definisanje podataka, azuriranje, kontrolu koinzistentnosti, konkurentan rad i jezik za odrzavanje recnika podataka. Ugradjen je u većinu komercijalno raspoloživih SUBP, od personalnih do server racunara. Osnovne karakteristike sql su:

- Jednostavnost i jednobraznost pri koriscenju(relacije se kreiraju jednom naredbom i odmah su spremene za koriscenje)
- Mogućnost interaktivnog i klasičnog programiranja (koristeći SQL dobijaju se odgovori na trenutno postavljene zahteve ili se SQL blokovi ugradjuju u neki viši programski jezik)
- Neproceduralnost(On je proceduralan u velikoj meri jer se njime definise STA se zeli a ne KAKO doci do rezultata)

Sql naredbe se mogu grupisati u vise grupa a mi ih mozemo podeliti na :

- Naredbe za definisanje podataka (Data Definition Statements – DDL) :  
CREATE ,ALTER,DROP...

-Naredbe za manipulisanje podacima (Data Manipulation Statements -  
DML): INSERT,SELECT,UPDATE,DELETE..

-Naredbe za kontrolne (upravljacke) funkcije (Data Control Statements):

GRANT(korisnicka prava),REVOKE(oduzma prava),COMMIT(prenos  
dejstva transakcije u bazu),ROLLBACK(ponistavanje transakcije)

## **18. Cilj relacione baze podataka:**

Cilj relacione baze podataka :

- Nezavisnosnost strukture podataka od programa koje ih koriste i obrnuto
- Minimalna redudansa
- Jednostavnije koriscenje i upravljanje podacima
- Koriscenje baze podataka od strane veceg broja korisnika

## **19. Anomalije odrzavanja podataka**

Da bi omogucili jednostavno koriscenje i menjanje podataka moramo spreciti anomalije odrzavanja podataka. U tu grupu spadaju:

- Anomalija dodavanja: Javlja se kada se informacije o svojstvima jednog objekta u bazi cuvaju kao deo opisa nekog drugog objekta.Npr. ako imamo podatke o katedri ili predmetu u samoj tabeli nastavnik to bi nam onemogucilo da te objekte dodamo osim ako ne postoji bar jedan nastavnik koji ce tu radili ili predavati taj predmet.
- Anomalija brisanja: to je inverzna anomalija dodavanja.Prime bi bio da ako u opisu nastavnika nalazimo svojstva predemtta na kojem predaje.Svakim brisanjem nastavnika brise se i ta kopija predmeta koja ce nam mozda nekada biti potrebna.
- Anomalija menjanja(Azuriranja): javlja se u slucaju kada promene podataka o jednom objektu treba izvorsiti na vise kopija objekata. Kao i sa proslim primerima ako postoji podatak o predmetu na kojem radi nastavnik u samom njegovom opisu onda pri izmeni tog predmeta

treba to izvršiti na onoliko mesta na koliko profesora predaje tu. Ako se to ne izvrši imamo više tvrdnji koje nisu istinite što predstavlja nekonzistentnu bazu podataka.

## **20. Normalizacija:**

Osnovni cilj relacionih baza podataka je da odgovarajuća baza podataka: Ne sadrži redundansu, Da se može jednostavno koristiti i menjati Formalne kriterijume prema kojima se utvrđuje da li model podataka ispunjava prethodne zahteve daju normalne forme. Normalizacija je proces provere uslova normalnih formi i po potrebi svodjenje šeme relacije na oblik koji zadovoljava iste. Očekuje se reverzibilnost tj da ne sme doći do gubitka informacija sadržanih u polaznoj relaciji i moguća je rekonstrukcija iste.

### **Metode normalizacije:**

Postoje dve tehnike normalizacije: vertikalna normalizacija i horizontalna.

Vertikalna normalizacija je proces u kome se nenormalizovana šema relacije transformise u skup manjih normalizovanih relacija. iz relacione šeme se izdvajaju obeležja koja stoje u nedozvoljenim odnosima sa ostalim obeležjima u šemi. Zasniva se na operacijama projekcije i skupa. Projekcijom relaciju vertikalno razbijamo na dve ili više manjih relacija dok nam prirodni spoj služi da bi dokazali reverzibilnost tj da bi rekonstruisali polaznu relaciju. Ima dve varijante: Normalizacija sintezom (ne izvodi se u koracima), Normalizacija dekompozicijom (izvodi se u koracima)

Horizontalna normalizacija je slučaj kada se relacija rastavlja na podskupove n-torki, odnosno fragmente relacije koji ispunjavaju određene uslove. Zasniva se na operacijama selekcije i unije ali sama tehnika je još uvek u razvoju.

### **Funkcionalna zavisnost obeležja:**

Funkcionalna zavisnost obeležja je ograničenje koje definiše vezu između obeležja gde jedno obeležje može tačno da odredi drugo obeležje. (Sva obeležja moraju biti zavisna od tačno jednog obeležja tj. On funkcionalno određuje ostala obeležja).

### **Tranzitivna zavisnost obelezja:**

To je ogranicenje gde se odredjuje funkcionalana zavisnost nekljucnih elemenata sa drugim ne kljucim elementima. Primer:

Razlog za postojanje opisanih anomalija u relaciji NFM je već navedena FZ  
NASTAVNIK → MESTO koju nazivamo i *tranzitivna zavisnost* jer proizilazi  
iz zavisnosti S\_NAS → FAKULTET, FAKULTET → MESTO.

### **Parcijalna zavisnost obelezja:**

Specijalan slucaj tranzitivne i vazi samo ako neklucna obelezja zavise od delova kljuka tj kada je primarni kljuc slozen kljuc.

### **Normalizacija dekompozicijom:**

Zapocinje od proizvoljne nenormalizovane relacione seme i izvodi se u koracima. Svakim korakom normalizacije sema se prevodi u visu normalnu formu tako sto polazni skup obelezja delimo u dva skupa obelezja i od svakog se formira posebna relaciona sema. Svaki od koraka mora biti reverzibilan.

### **Uslovi dekompozicije bez gubitka informacija:**

Generalno gledano kod dekompozicije se po pravilu ne gube podaci vec ih po spajanju ima vise nego sto je bilo u polaznoj relaciji. Uslovi koje bi morali da ispunimo da dekompozicija bude bez gubitaka bi bili:

- Novonastala sema relacije mora biti podskup pocetne seme relacija i da unija tih relacija predstavlja skup obelezja pocetne seme.
- Da bi dekompozicija bila reverzivna presek projekcija dve manje seme relacija koje su nastale od pocetne ne sme biti prazan.
- Skup obelezja koja cine presek manjih relacija moraju biti kandidati za kluc u bar jednoj od relacija

### **Vertikalna normalizacija dekompozicijom:**

U kontekstu vertikalne normalizacije definisemo 6 normalnih formi:

- Prva normalna forma (1NF)
- Druga normalna forma (2NF)
- Treća normalna forma (3NF)
- Codd-ova normalna forma (BCNF)
- Četvrta normalna forma
- Peta normalna forma

Kod postupka normalizacije semu prvo transformisemo u u 1NF pa zatim u 2NF,3NF i tako redom.Sto je redno broj forme veci to su i njeni uslovi stroziji i broj funkcionalnih zavisnosti se smanjuje.

### Uslovi ispunjenosti 1NF:

Sema relacije je u prvoj normalnoj fazi ako i samo ako je domen svakog od njenih obelezja skup atomiranih vrednosti i sva nekljucna obelezja nisu funkcionalno zavisna od primarnog kljuca.S obzirom da u kontekstu modela relaciju definisemo kao neprazan skup dekartovog proizvoda atomiranih vrednosti sledi onda da je savaka sema u 1NF.

### Uslovi ispunjenosti 2NF:

Relaciona sema se nalazi u 2NF ako je svako nekljucno obelezje potpuno funkcionalno zavisno od kljuca relacije.To znaci da ne sme postojati parcijalna zavisnost i svi elementi moraju zavisiti od celog kljuca umesto da podatak mozemo definisati i samo delom kljuca.

### Uslovi ispunjenosti 3NF:

Relaciona sema R nalazi se u 3NF ako je u 2NF i ako nijedno nekljucno obelezje u R nije tranzitivno zavisno od primarnog kljuca od R vec cu svi funkcionalno zavisni od njega.

### Postupak svodjenja seme relacije koja nije u 2NF u skup seme relacije koje su u 2NF:

Ako imamo tabelu kao u primeru dole koja ima identifikator i svaka kolona ima atomirane vrednosti tj. Svaka celija ima istu vrednost i svaka kolona ima isti tip vrednosti. Onda pokusavamo da je dekomponujemo na vise podtabela koje ce zadovoljiti uslov 2NF a to je da sva nekljucna obelezja moraju zavisiti od primarnog kljuca te tabele

Primary Key

Cust ID	Cust Name	Item	Shipping Address	Newsletter	Supplier	Supplier Phone	Price
at_smith	Alan Smith	Xbox One	35 Palm St, Miami	Xbox News	Microsoft	(800) BUY-XBOX	250
roger25	Roger Banks	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300
wilson44	Evan Wilson	Xbox One	28 Rock Av, Denver	Xbox News	Microsoft	(800) BUY-XBOX	250
wilson44	Evan Wilson	PS Vita	28 Rock Av, Denver	PlayStation News	Sony	(800) BUY-SONY	200
am_smith	Alan Smith	PlayStation 4	47 Campus Rd, Boston	PlayStation News	Sony	(800) BUY-SONY	300

Nakon toga dobijamo dve tabele koje izgledaju ovako:

Primary Key

Cust ID	Cust Name	Shipping Address	Newsletter
at_smith	Alan Smith	35 Palm St, Miami	Xbox News
roger25	Roger Banks	47 Campus Rd, Boston	PlayStation News
wilson44	Evan Wilson	28 Rock Av, Denver	Xbox News
wilson44	Evan Wilson	28 Rock Av, Denver	PlayStation News
am_smith	Alan Smith	47 Campus Rd, Boston	PlayStation News

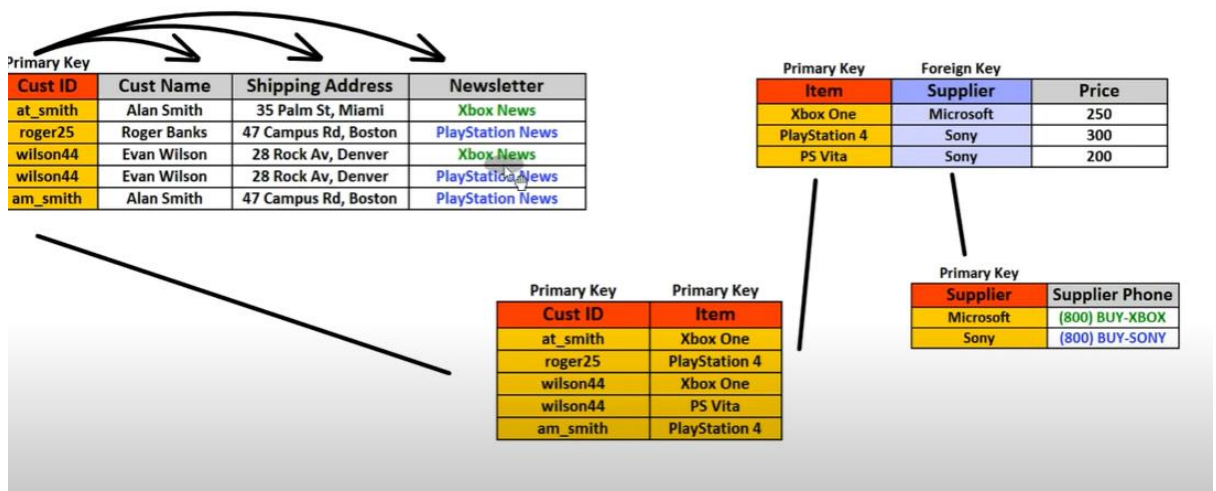
Primary Key

Item	Supplier	Supplier Phone	Price
Xbox One	Microsoft	(800) BUY-XBOX	250
PlayStation 4	Sony	(800) BUY-SONY	300
PS Vita	Sony	(800) BUY-SONY	200

Primary Key	Primary Key
Cust ID	Item
at_smith	Xbox One
roger25	PlayStation 4
wilson44	Xbox One
wilson44	PS Vita
am_smith	PlayStation 4

Da bi proverili da li je doslo do gubitka mozemo uraditi uniju obelezja obe tabele i proveriti da li se poklapa sa pocetnom tabelom. Takodje mozemo da uradimo presek ili razliku kljucnih obelezja dve tabele i kao rezultat moramo dobiti kljucna obelezja one trece.

Zatim da bi ova tabela bila u 3NF moramo ukloniti tranzitivnu zavisnost tj moramo ukloniti sve nekljucana obelezja koja nisu funkcionalno zavisna od primarnog kljucja. Tada mozemo dobiti tabele kao sa slike:



## **21. Prevodjenje MOV u relacioni model podataka:**

### **Zasto se vrši prevodjenje:**

S obzirom da za sada ne postoji komercijalno raspoloživ SUBP zasnovan na MOV to znači da je praksi potrebno prevesti model radjen po MOV u neki drugi model podataka (najcesce relacioni). Kod prevodjenja se može zapaziti da svakoj apstraktnoj operaciji bogatog modela (3 nivoa) odgovara procedura semanticki manje bogatog modela (2 nivoa relacioni). Ovaj postupak se može automatizovati uz pomoc softverskih alata za prevodjenje MOV-a u relacioni model.

### **Opšta pravila:**

Opšta pravila iz kojih se izvode sva pojedinačna su:

Obezbediti jedinstvenost ključa,  
Izbeći višeznačna obeležja,  
Izbegavati neprimenljiva svojstva,  
Obezbediti minimalan skup šema relacija.

### **Pojedinacna pravila za objekte:**

Pravila za glase:

- Svaki objekat iz MOV postaje šema relacije. Ime tipa objekta postaje ime šeme relacije. Obeležja tipa su obeležja šema relacija. Za osnovne objekte identifikator objekata postaje primarni ključ šeme relacije.
- Identifikator nadređenog objekta postaje jedno od obeležja šeme relacije koja odgovara slabom objektu, i postaje deo ključa šeme relacije slabog objekta, zajedno sa obeležjima koja jednoznačno identifikuju pojavljivanje "slabog" objekta u okviru pojavljivanja njemu nadređenog objekta.
- Nadtip i podtip takodje postaju šema relacije. Ime šeme relacije i obeležja jednaka su imenu objekta i obeležjima samog objekta.

### **Pojedinacna pravila za veze:**

Pravila za pojedinačne binarne veze glase:

- **Veza 1:1** (Obavezna sa obe strane) :  
Sama veza i oba objekta koji u njoj učestvuju postaju jedna šema relacije.  
Obeležja šeme relacije su svi atributi jednog i drugog objekta i atributi veze (odnosno agregacije).



Kandidati za ključ u ovoj šemi relacije su identifikatori i jednog i drugog objekta koji su vezi.

- **Veza 1:1**(Sa jedne strane obavezna):

Dva objekta u vezi i sama veza daju dve seme relacije s tim što se identifikator jednog od objekata koji su u vezi uvrsti u obeležja druge seme relacije .

Veza se predstavlja spoljnim ključem u onoj šemi relacije u kojoj vrednost tog obeležja mora biti data.

- **Veza 1:1**(Opciona sa obe strane):

Kreiraju se tri šeme relacija.

Obeležja u semi relacije koja odgovara vezi su i identifikatori objekata koji su u vezi i oba su kandidati za ključ.

Za primarni ključ se bira jedan od identifikatora objekata koji učestvuju u vezi

- **Veza 1:M**(Sa jedne strane obavezna):

Ne postaje posebna sema relacije već identifikator objekta za koji je  $GG=M$  postaje obeležje šeme relacije koja odgovara objektu sa strane za koju je  $GG=1$ .

Veza između jakog i slabog objekta kao i veza između podtipa i nadtipa ne postaje posebna sema relacije.

- **Veza 1:M**(Sa jedne strane opciona):

Postaje posebna sema relacije

Obeležja te seme relacije su identifikatori objekata koji su u vezi, a ključ seme relacije je identifikator objekta za koji je  $GG=1$

Agregacija mesovit tip objekt veze se prevodi kao odgovarajući tip veze, a obeležje agregacije ide u semu relacije kojom se rešava veza.

## 22. Osnove obrade transakcije:

### Sta je transakcija:

Transakcija je vremenski uredjeni niz nedeljivih radnji nad bazom podataka koje u celini ne remete uslove integriteta. Predstavljaju neku logicku jedinicu rada nad bazom podataka. Ona transformise bazu iz jednog koinzistentnog stanja u drugo koinzistentno stanje. Bitno je da transakcija bude tako oformljena da usled njenog prekida ne dodje do promene same baze podataka.

**Sql naredbe za rukavanje transakcijama:**

Commit(signalizira o uspesnom zavrsetku transakcije, gde se onda dejstva transakcije mogu sacuvati i baza ostaje u koinzistentnom stanju)  
Rollback(signalizira da je transakcija neuspesno izvršena i u ovom slucaju treba ponisiti njeno dejstvo)

**Paralelno izvršavanje transakcija:**

Sistem za upravljanje bazom podataka treba da omoguci paralelan rad vecem broju korisnika nad istim podacima sto znaci paralelno izvršavanje vise transakcija odjednom. Da bi to bilo moguće mora se obezbediti neko adekvatno upravljanje izvršavanjem transakcija kako one ne bi nepoželjno mesale svoja dejstva jedna na drugu. Zbog mesanja transakcija može nastati pogresan rezultat jer pri mesanju operacija iz dve korektne transakcije može nastati nekorektan rezultata.

Ti problemi mogu biti:

- Problem izgubljenog azuriranja
- Problem zavisnosti od privremenih azuriranja
- Narusavanja serijabilnosti

**Problem izgubljenog azuriranja:**

Ako bi uzeli u obzir dve transakcije A i B. Transakcija A cita zapis u jednom vremenskom trenutku dok B u drugom. A ce onda nakon toga u odredjenom trenutku azurirati podatke na osnovu onih procitanih na pocetku dok ce be to isto uratiti ali u malo kasnijem vremenskom intervalu cime se delovanje transakcije A ponistava jer B u trenutku svog izvršavanja nije znala za nju i njeno dejstvo.

**Problem pri zavisnosti od privremenog azuriranja:**

To je slucaj kada bi na primer kada bi transakciji A dozvolili citanje a jos gore i azuriranje zapisa koji su privremeno azurirani od transakcije B pri cemu se taj zapis jos nije trajno upisao u bazu. Ako u nekom momentu transakcija B ponisiti svoje privremena azuriranja dolazimo u situaciju gde je transakcija A izvršila operacija nad bazom sa podacima koji nikada nisu ni postojali u njoj.

**Osobine transakcija su:**

Koinsitencija

Izolacija

Atomicnost

Trajnost

**Protokoli zaključavanja:**

Uobicajeni način resavanja problema pralelnog izvršavanja jesu protokoli zaključavanja. Osnovna ideja zaključavanja jeste u tome da transakcija koja zahteva garancije da neki objekta za koji je zainteresovana ne može biti ažuriran u nekom vremenu. Takav objekat se zaključava i sprečava se njegovo ažuriranje od drugih transakcija. Na taj način transakcija raspolaze sa stabilnim podacima o objektu koliko to ona zahteva. Zaključavanja može biti direktno (sami navodimo) i medjusobno.

**Restauracija konzistentnog stanja baze podataka:**

Administrator transakcija u sebi ima implementiran protokol nedeljivog kompletiranja transakcije kojim se sva ažuriranja BP upisuju u log transakcije. Mehanizam upisa u log transakcije može biti:

- Upis novih vrednosti BP u log transakcije (after image)
- Upis starih vrednosti BP u log transakcije (before image)

Svaka tačka sinhronizacije se unosi log transakcije. Logovi pojedinačnih transakcija se kombinuju u jedinstven sistemski log koji nazivamo dnevnik ažuriranja, u kojem su pokazivačima povezani zapisi koji predstavljaju log jedne transakcije.

Dnevnik ažuriranja omogućuje oporavak BP u kombinaciji sa sigurnosnom kopijom BP.