

Visoka škola
elektrotehnike i računarstva strukovnih studija
Beograd

BAZE PODATAKA

Uvod u baze podataka

SADRŽAJ:

- Uvod**

- Pojam baze podataka (BP)**

- Informacioni sistemi (IS)**

 - Informacija i podatak

 - Životni ciklus IS

- Strukture podataka**

- Modeli podataka**

- Programi za rad sa BP**

 - DBMS

 - Klasični sistemi i BP

- Relacione BP**

- Entitet, Instanca, Atribut i identifikator**

- Modelovanje BP**

Uvod

- Težnja da se različite vrste softvera (aplikacija) približe analogiji sa realnim svetom
 - Windows, Word, Excel, ...
- Baze podataka – potpuno veštačka tvorevina
 - Slično geometriji
- Baze podataka nisu tabele
 - Tabele postoje u realnom svetu (telefonski imenik, rečnik i sl)
 - Relacione baze podataka ne postoje u realnom svetu

- Baze podataka u elektronskoj formi: telefonski imenici, liste radnika, proizvodi, časopisi, rezervacije avionskih karata itd.
- Različitim manipulacijama nad podacima koji su smešteni u bazi dobijamo dragocene informacije.

Prava snaga baze podataka ne leži u mogućnosti pohranjivanja informacija, već u **sposobnosti pronalaženja i preuzimanja tačno onih informacija koje želimo.**

Šta su baze podataka?

- Podrazumeva set podataka smeštenih u računaru
- BP postoje u različitim formama:
 - Elektronski podaci: tekst, video i audio zapisi, brojevi
 - Knjige: kolekcija tekstova i slika, biblioteke
 - Datoteke organizovane u direktorijume
 - Skup šematskih crteža
 - Uzorci sa naftnih bušotina
 - Uzorci krvi za medicinske laboratorije
 - DNK uzorci u forenzičkim laboratorijama
 - Geografski podaci
 - Itd..

Baza podataka predstavlja organizovan skup logički povezanih podataka koji su smešteni u jednoj ili više datoteka, tako da se mogu pretraživati, sortirati, ažurirati, grupisati i štampati.

Definicije

- **Baza podataka**
 - **organizovani skup logički povezanih podataka**
 - integrisani skup podataka o nekom sistemu i skup postupaka za njihovo održavanje i korišćenje, organizovan prema potrebama korisnika
 - **Baza podataka (*database*)** je kolekcija međusobno povezanih podataka, uskladištenih sa minimumom redundanse, organizovanih tako da u najkraćem vremenu daju informacije

Podaci na računarima

- Sastoje se od:
 - **Bit**a, organizovanih u bajtove, koji mogu da čine stringove itd.
 - **Polja** (*fields*) – niz bajtova koji reprezentuje informaciju (broj, tekst, slika, audio/video zapis, ...)
 - **Zapisa** (*records*) - niz polja koji zajedno opisuju jedan “entitet”
 - **Datoteka** (*files*) – niz zapisa koji opisuju različite entitete
- Baze podataka sadrže više relevantnih datoteka

INFORMACIONI SISTEMI

Informacija i podatak

Informacija $\neq$ podatak

- Pod **informacijom** se podrazumeva svaki smisaoni iskaz koji primaocu donosi neku novost. Svaka **nova informacija uvećava znanje primaoca o ljudima, stvarima ili događajima, i u isto vreme smanjuje neizvesnost o nekom pojmu.**

- Primeri

MARKO JE VISOK 180 CM.

IVANA PETROVIĆ JE STUDENT III GOD. VIŠER U BEOGRADU

MARKOV AUTOMOBIL JE CRVENE BOJE.

Podaci

- Podatak
 - **Činjenica** o nekom predmetu i/ili događaju koja se može zabeležiti i sačuvati na računaru
 - **Strukturirani podaci**: brojevi, karakteri, ...
 - Nestrukturirani podaci: slika, zvuk, video, ...
 - Podatak sam po sebi nema značenje, tek kada se interpretira nekom vrstom sistema za obradu podataka poprima značenje i postaje **informacija**.

Pojam podatka

Spoznajom svojstva pojedinih pojava i objekata ili
Ishoda događaja, **smanjuje je stepen neodređenosti** sveta
koji nas okružuje:

- DOBILI SMO NEKU INFORMACIJU

Prikazivanjem informacija u formalizovanom obliku
pogodnom za

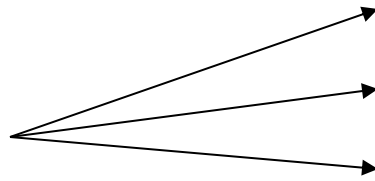
**ZAPISIVANJE, PRENOS, SKLADIŠTENJE i OBRADU –
nastaju PODACI**

(Podaci su diskretni zapisani fakti o fenomenima i iz njih
se dobijaju Informacije o svetu)

- **Informacija**

- Podatak koji je obrađen na takav način da se znanje osobe koja koristi podatak povećalo

Podaci koji nisu korisni



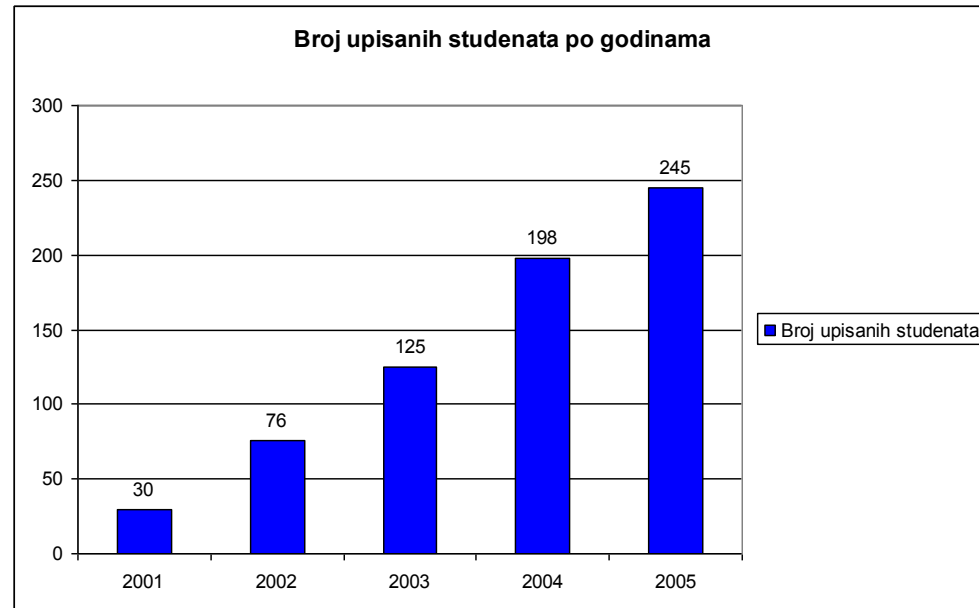
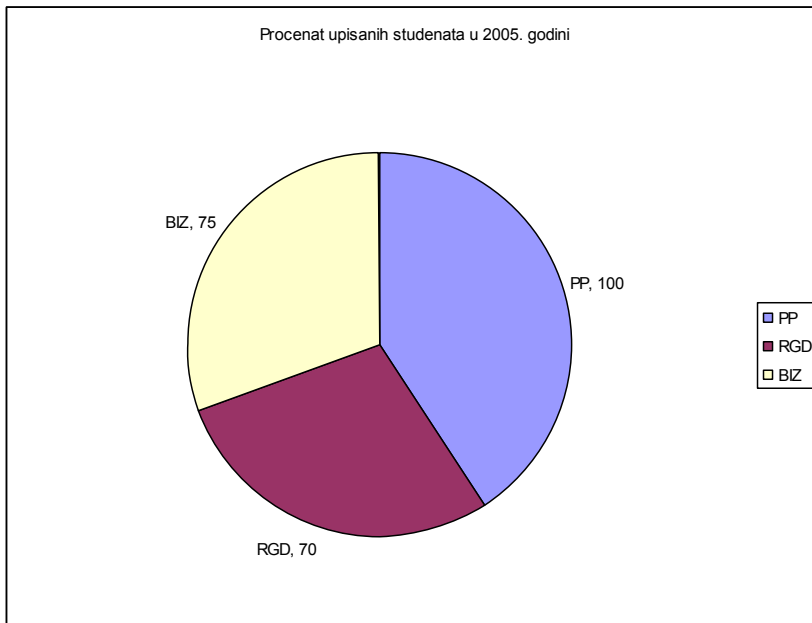
Petar Petrović	1506983710325
Marko Marković	0211979850123
Janko Janković	1112985830456
- - - - -	- - - - -

Ime i prezime	JMBG	Smer	Godina upisa
Petar Petrović	1506983710325	IS	2017
Marko Marković	0211979850123	RT	2017
Janko Janković	1112985830456	IS	2016
- - - - -	- - - - -	IS	2016

Informacija o upisu – dopunjeni relevantni podaci

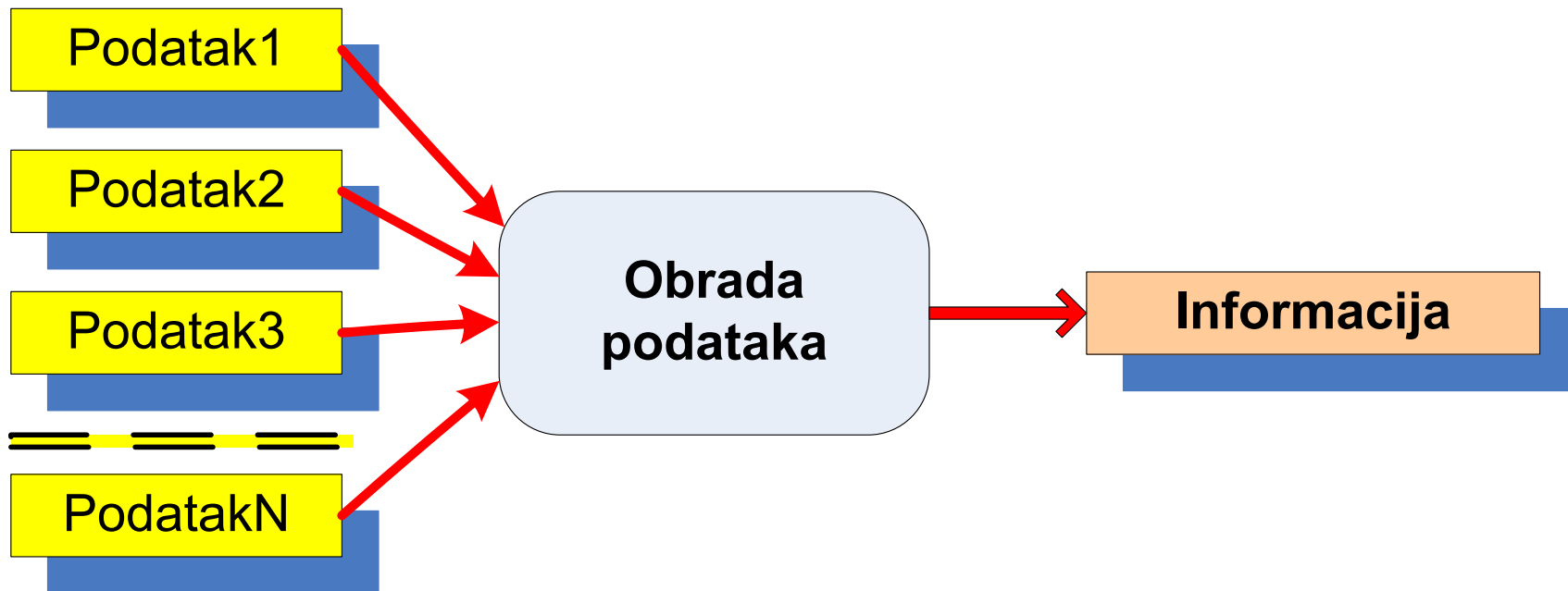
- **Informacije**

- podaci koji se sumiraju ili na neki drugi način obrađuju i prezentuju



Grafički prikaz podataka iz BP - informacija o upisu

Informacija

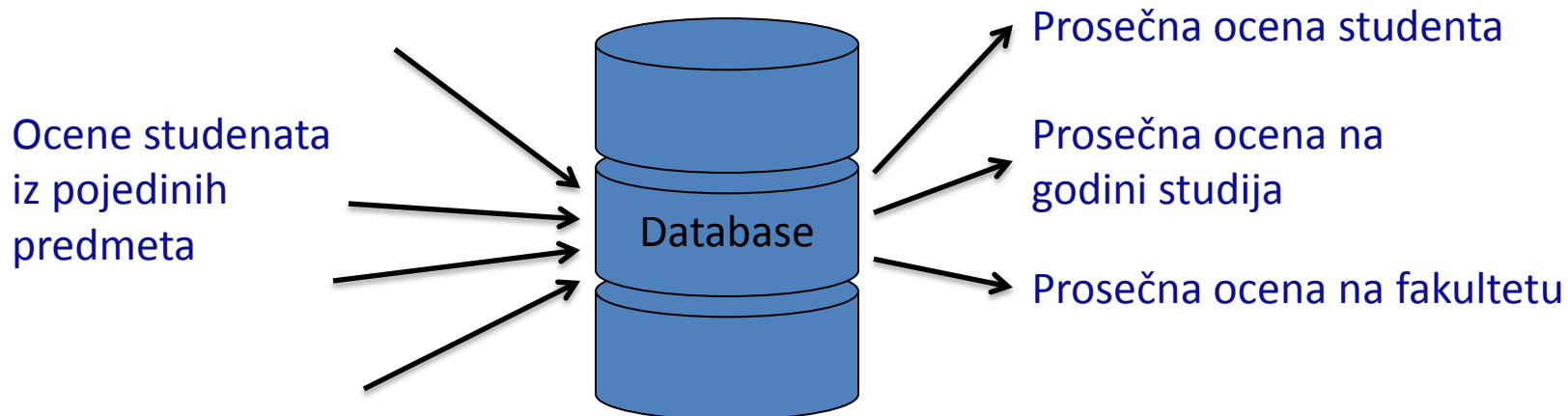


Važno je razumeti šta je smešteno i čuva se u BP, a šta se može dobiti iz BP

Podaci/Informacije

PODACI

INFORMACIJE



Obrada podataka

PRIKUPLJANJE (zapis i prenos)

ORGANIZOVANJE

SKLADIŠTENJE (MEMORISANJE)

OBRADA

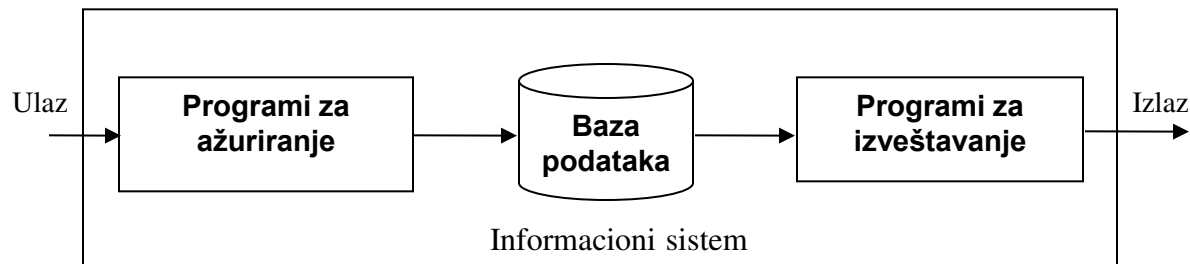
PREZENTOVANJE

REZIME

Podaci/Informacije

- **Baza podataka** je organizovana kolekcija međusobno povezanih podataka
- **Podaci** predstavljaju činjenice koje se mogu sačuvati i imaju neko implicitno značenje
- Korisnici mogu da **pregledaju, pretražuju, brišu, dopunjuju i ispravljaju podatke**

INFORMACIONI SISTEMI



Osnovna struktura informacionog sistema

Većina informacionih sistema se kreira tako da se uklopi u postojeću organizaciju, mada veliki broj novih informacionih sistema za svoju implementaciju zahteva i reorganizaciju postojećeg poslovnog sistema

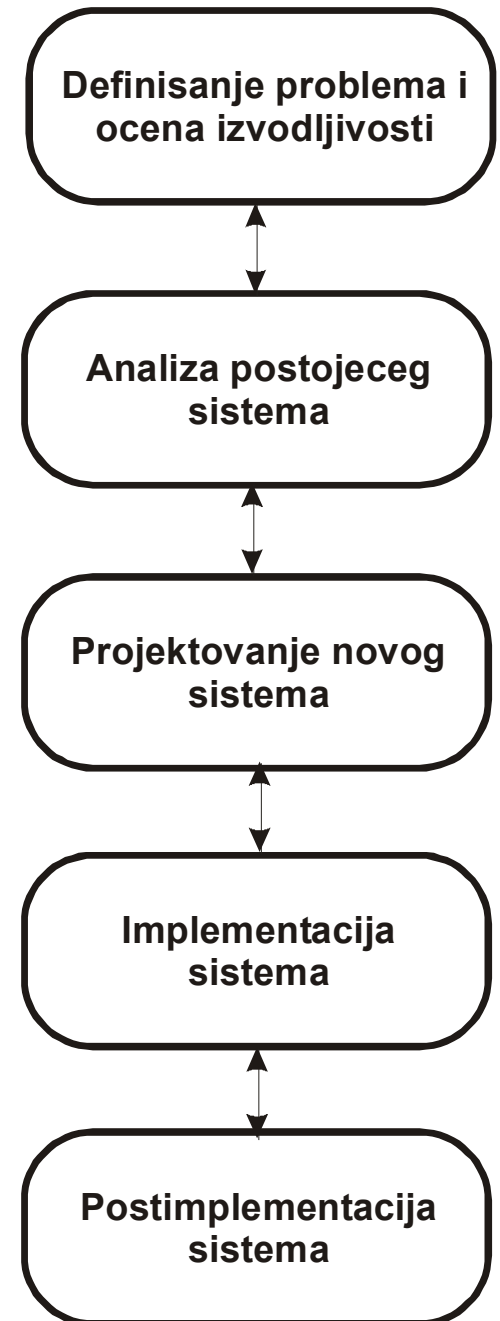
- Obzirom da je razvoj informacionih sistema složen posao koji nije uvek lako isplanirati, čest je slučaj da se projekat prekine kada je izvesno da se ne može okončati na vreme i sa planiranim sredstvima.
- Oblast koja se bavi razvojem informacionih sistema se naziva **sistemska analiza**
- Ljudi koji rade na razvoju informacionih sistema se nazivaju **sistem analitičari**.
 - Sistem analitičari, na osnovu razgovora sa korisnicima i upravom određene organizacije, moraju razumeti misiju organizacije i njene strateške ciljeve. Njihov zadatak je da definišu zahteve neophodne za modifikaciju postojećeg sistema ili izradu novog, da identifikuju i procene alternativna rešenja, razgovaraju sa korisnicima i dokumentuju ceo proces, sa objašnjenjima kako i zbog čega su izvršene određene akcije.

Proces nastanka i rada informacionog sistema
se naziva

Životni ciklus informacionog sistema

Životni ciklus informacionih sistema

- *Analiza postojećeg sistema (System Analysis)*
 - postojeći fizički model
 - “Klijent šalje narudžbenicu poštom ili telefonom” itd.
 - postojeći logički model
- *Projektovanje novog sistema (System Design)*
 - novi logički model
 - novi fizički model
- *Implementacija sistema*
 - Na više načina:
 - paralelnim radom starog i novog sistema jedno kraće vreme (najsigurniji način),
 - korišćenje novog sistema od strane jednog dela organizacije, dok drugi koriste i dalje stari sistem,
 - uvođenje novog sistema po fazama,
 - direktna zamena koja podrazumeva potpunu zamenu starog sistema novim, što može biti riskantno u nekim slučajevima.



STRUKTURE PODATAKA

Strukture podataka

- Struktura podataka **definiše veze među podacima**.
- Postoje dve različite strukture podataka: **logička** i **fizička struktura**.
- **Logička struktura podataka** prikazuje **logičke veze među podacima, onako kako su definisane i u realnom svetu, bez obzira na način smeštanja podataka na fizičke medijume**.
- **Fizička struktura podataka** predstavlja **način smeštanja podataka na fizičke medijume**, što je uslovljeno logičkom strukturom podataka i fizičkim karakteristikama medija na koji se podaci zapisuju.

Strukture podataka

- U logičkoj strukturi podataka, najmanja jedinica podataka je **elementarni podatak (data)** i smatra se nedeljivom.
- Elementarni podatak može biti predstavljen **različitim tipovima podataka**.
- Najčešći tipovi podataka su **tekstualni, numerički (sa različitim formatima zapisa), datumski, logički i drugi**.
- Odgovarajuća jedinica podataka u fizičkoj strukturi je **polje (field)** Polje je posebno područje na fizičkom medijumu koje se koristi za smeštaj jednog tipa podataka. Za polje se definiše veličina polja. Na primer, polje u kome će biti smeštena oznaka raznih država može biti veličine tri karaktera.

Strukture podataka

- **Slog (record)** je skup povezanih elementarnih podataka koji čine celinu, pa se može govoriti npr. o slogu podataka o jednom studentu. Razlikujemo **logički** i **fizički** slog.
- **Logički slog** je skup logički povezanih podataka, recimo slog podataka o jednom studentu.
- **Fizički slog** je slog posmatran sa stanovišta njegove fizičke pozicije na medijumu.
- Fizički slogovi se mogu grupisati u veću celinu koja se naziva **blok (block)**. Blok je niz elemenata koji se zapisuju ili prenose kao celina.

- **Baza podataka (database)** se sastoji od jedne ili više povezanih datoteka.
- Baza podataka predstavlja skup povezanih podataka pogodnih za **dodavanje, skladištenje i izdavanje podataka za više nezavisnih korisnika.**
- Recimo, može se govoriti o bazi podataka jedne škole, preduzeća, organizacije i drugo. **Baza podataka podrazumeva i fizičku i logičku strukturu podataka.**

- **Šema baze podataka** je opis strukture baze podataka na različitim nivoima posmatranja.
- **Podšema baze podataka** je deo šeme baze podataka i predstavlja pogled na bazu podataka iz ugla jedne aplikacije, odnosno, jednog korisnika.

Metapodaci

- **Metapodaci**

- Rečnik podataka
- **Podaci o podacima**
- Opisuju svojstva ili karakteristike podataka krajnjih korisnika i kontekst tih podataka
- Tipična **svojstva podataka su naziv (ime) podatka, definicija, dužina (veličina), i dozvoljene vrednosti**
- Metapodaci **omogućavaju dizajnerima i korisnicima baza podataka da razumeju koji podaci postoje u bazi i šta oni znače**

MODELI PODATAKA

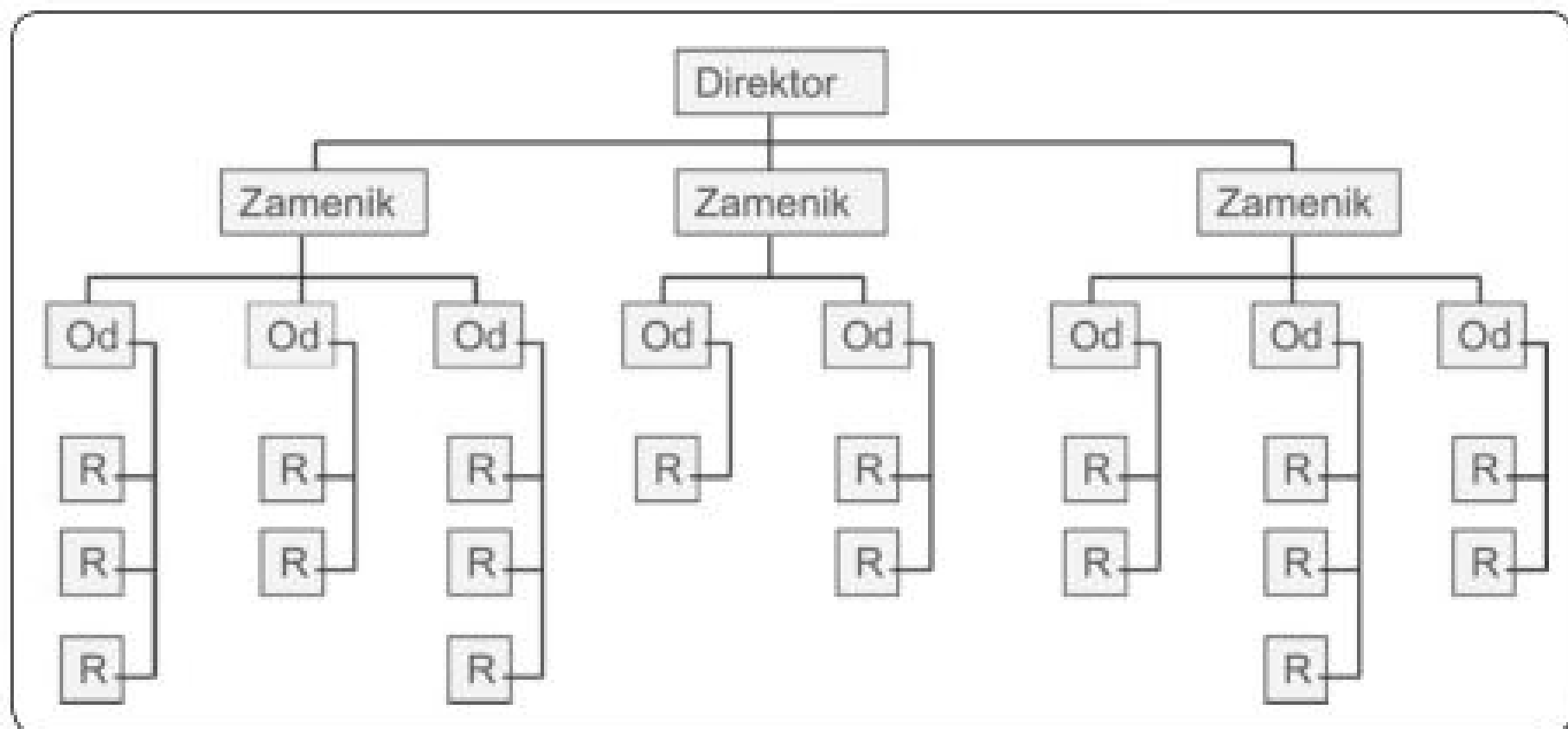
Modeli podataka

- Baza podataka preko svoje strukture odslikava stanje jednog dela realnog sveta.
- Da bi se slika realnog sveta adekvatno prikazala u bazi podataka, potrebno je izgraditi **model**.
- **Model podataka** treba da prikaže objekte realnog sveta i njihove međusobne odnose.

Modeli podataka

- Podaci su logički organizovani po nekom modelu
- Model čini osnovu za osmišljavanje, definisanje i implementaciju baze podataka
- Osnovni modeli baza podataka:
 - Hijerarhijski model
 - Mrežni model
 - Relacioni model
 - Objektno-orijentisani model

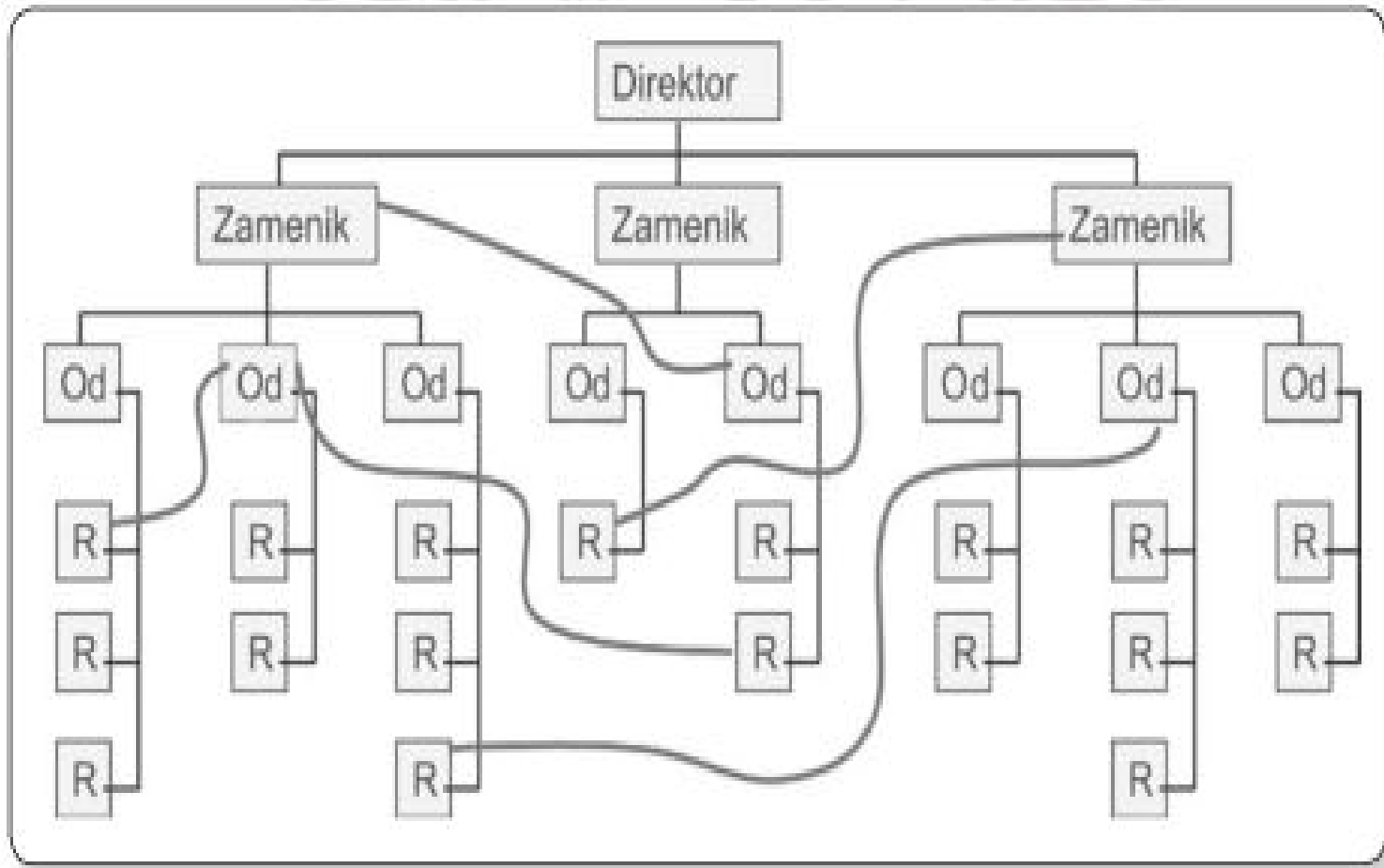
Hijerarhijski model



Hijerarhijski model

- Prednosti
 - Predstavlja pogodan opis realnog sistema
 - Pogodan je za programere (struktura stabla)
 - Slogu na nižem nivou moguće je pristupiti samo preko svih slogova viših nivoa
- Nedostaci
 - Može se dodati samo onaj slog koji ima roditelja (anomalija dodavanja)
 - Brisanjem roditelja brišu se i svi potčinjeni slogovi (anomalija brisanja)
 - Mogućnost postojanja samo jednog roditelja

Mrežni model



Mrežni model

- Prednosti
 - Podrška vezama M:M
 - Konceptualna jednostavnost
 - Nezavisnost podataka
 - Fleksibilan pristup podacima
- Nedostaci
 - Složenost sistema (problem skalabilnosti)
 - Optimizacija - ručna

Relacioni model

- Korisnik ga vidi kao skup tabela za skladištenje podataka
- Tabele su predstavljene kao presek redova i kolona
- Tabele su povezane na osnovu zajedničkih karakteristika
- E. F. Codd 1970. (pravila normalizacije)

Relacioni model

Table name: AGENT

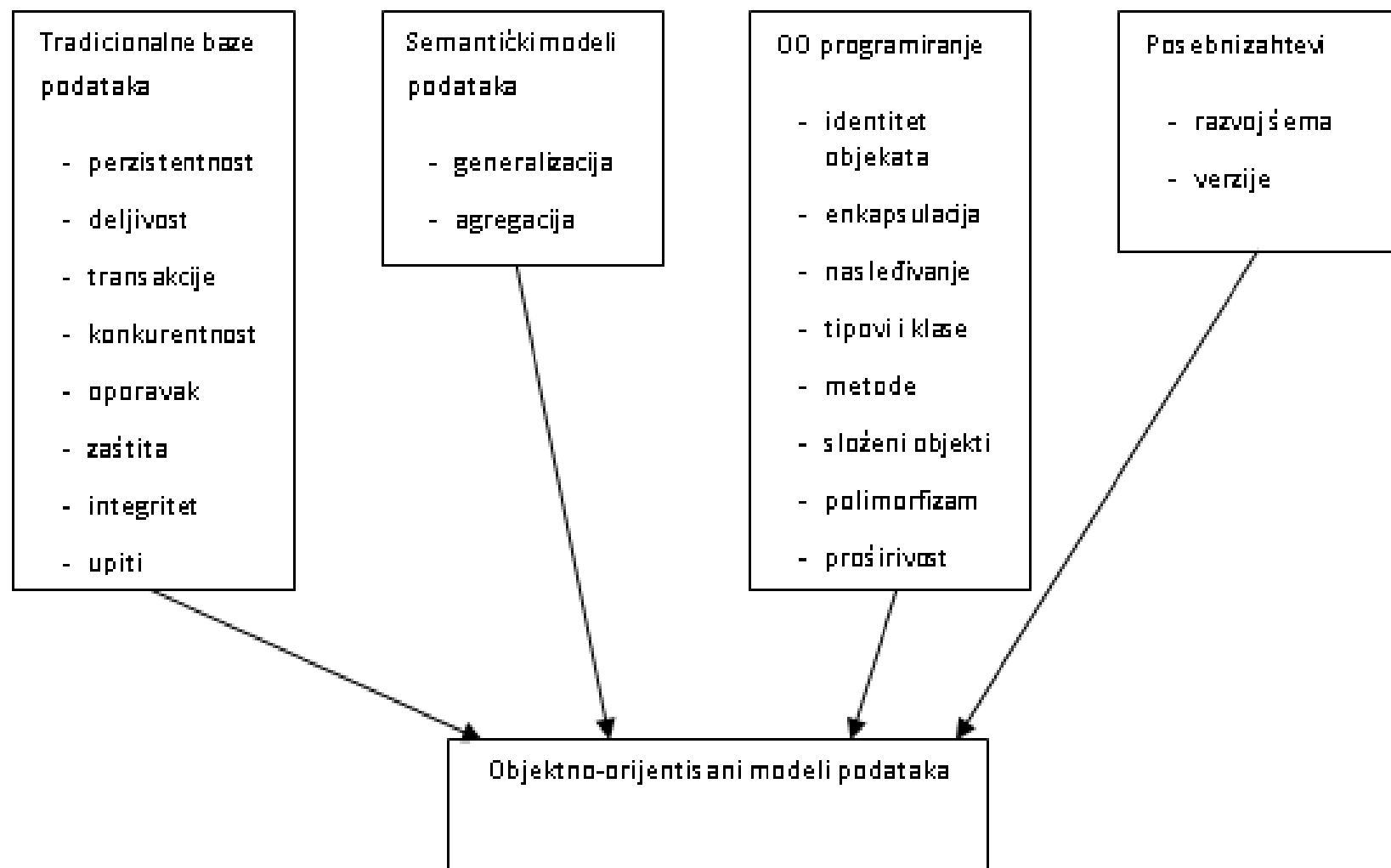
	AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
►	501	Alby	Alex	B	713	228-1249
	502	Hahn	Leah	F	615	882-1244
	503	Okon	John	T	615	123-5589

Link through AGENT code

Table name: CUSTOMER

	CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
►	10010	Ramas	Alfred	A	615	844-2573	05-Apr-2002	502
	10011	Dunne	Leona	K	713	894-1238	18-Jun-2002	501
	10012	Smith	Kathy	W	615	894-2285	29-Jan-2001	502
	10013	Olowski	Paul	F	615	894-2180	14-Oct-2002	502
	10014	Orlando	Myron		615	222-1672	28-Dec-2002	501
	10015	O'Brian	Amy	B	713	442-3381	22-Sep-2002	503
	10016	Brown	James	G	615	297-1228	25-Mar-2002	502
	10017	Williams	George		615	290-2556	17-Jul-2002	503
	10018	Farriss	Anne	G	713	382-7185	03-Dec-2002	501
	10019	Smith	Olette	K	615	297-3809	14-Mar-2002	503

Objektno-orijentisani model



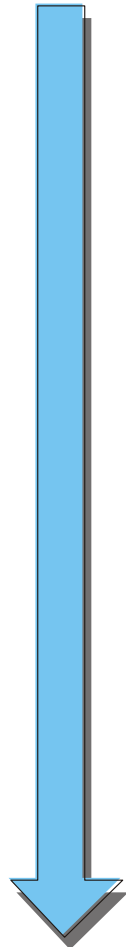
Objektno-orijentisani model

- Prednosti
 - Podrška bogatijem skupu tipova podataka i pravila
 - Skalabilnost (proširivost)
 - Nasleđivanje atributa (generalizacija)
 - Pristup podacima ograničen definisanim operacijama
- Nedostaci
 - Nepostojanje univerzalnog modela
 - Nedostatak iskustva i standarda

EVOLUCIJA MODELA PODATAKA

SEMANTIKA

najmanje



najviše

Hijerarhijski

Teskoća predstavljanja M:N odnosa

Mrežni

Fizički nivo zavisnosti

Relacioni

Slab semantički sadržaj

Entitet - veze

Lako za razumevanje
Semantički bogat

Semanticki

Više semantike u modelu
Podrzava nasledjivanje

Objektno
orjentisani

Prosireni
relacioni model

PROGRAMI ZA RAD SA BAZAMA PODATAKA

Programi za rad sa bazama podataka

- datotečni ili fajl sistemi

(file management programs) – KLASIČNI SISTEMI

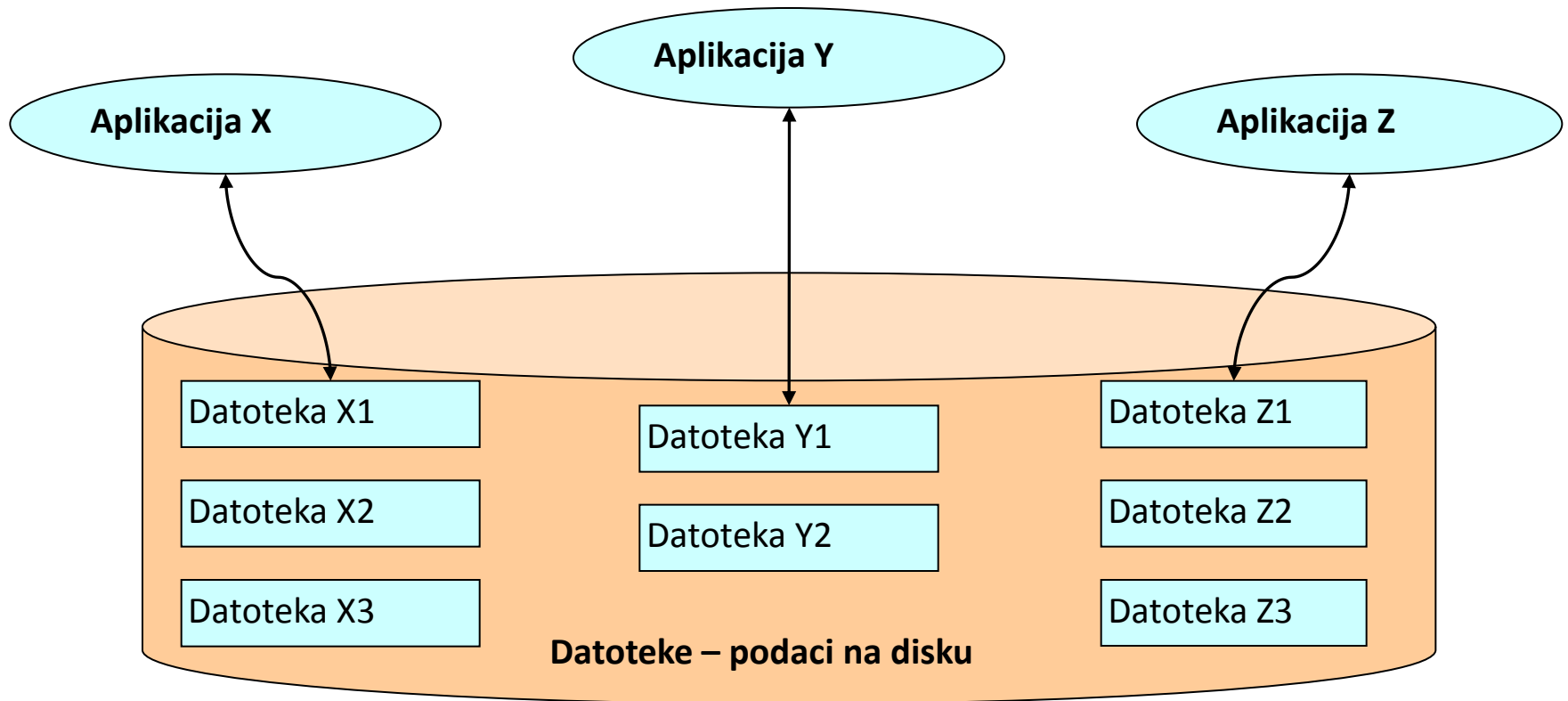
- sistemi za upravljanje bazama podataka (*database management systems - DBMS*)

Klasični sistemi i baze podataka

Klasični sistemi

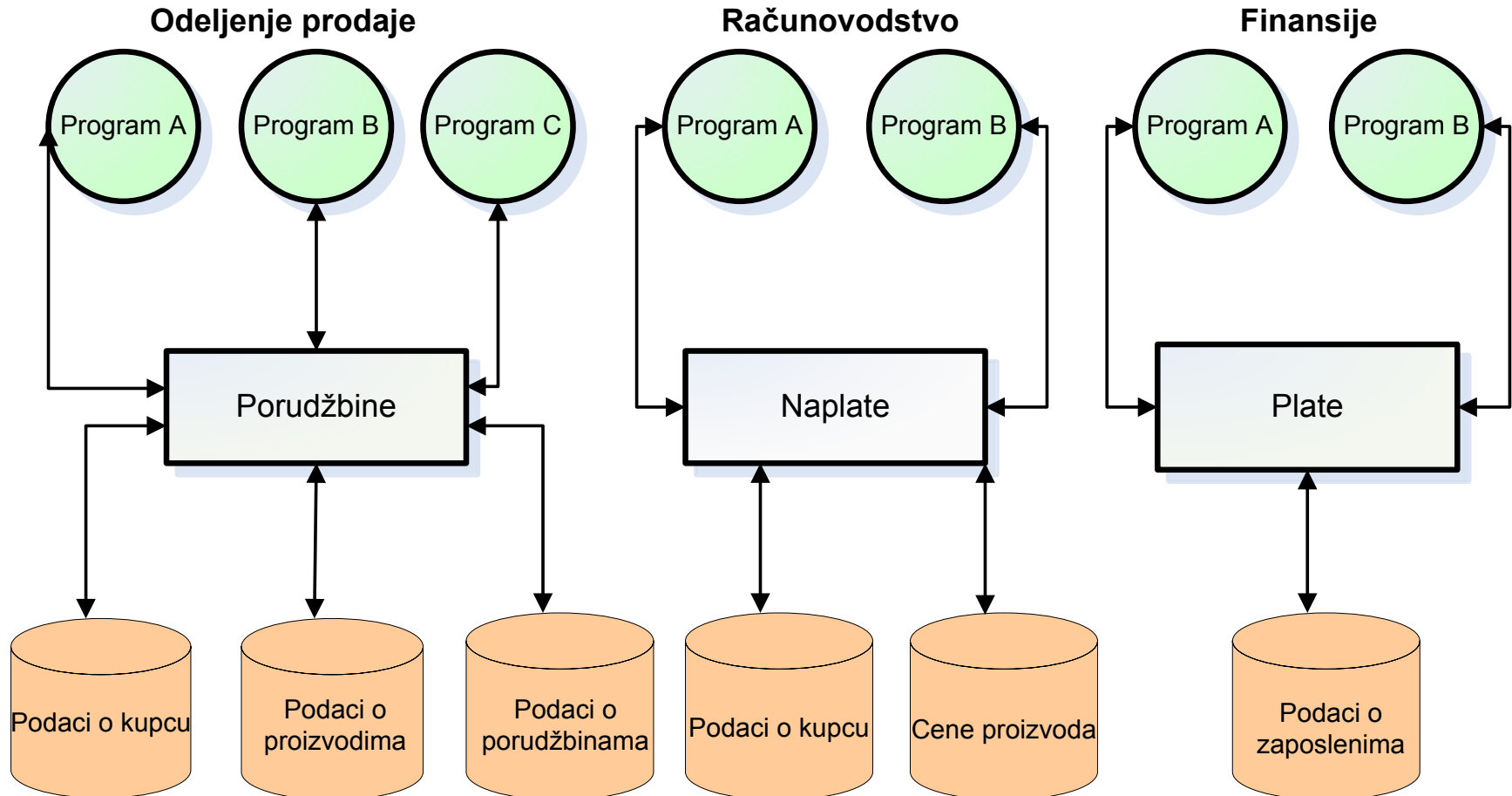
- Početak primena računara za obradu podataka
 - Nisu postojale BP
- Klasični sistemi – datoteke i programski jezici
- Kompleksnije poslovne aplikacije
 - Brojne mane sistema zasnovanih na datotekama i programskim jezicima

Klasični sistemi



Klasični sistemi

- primer -



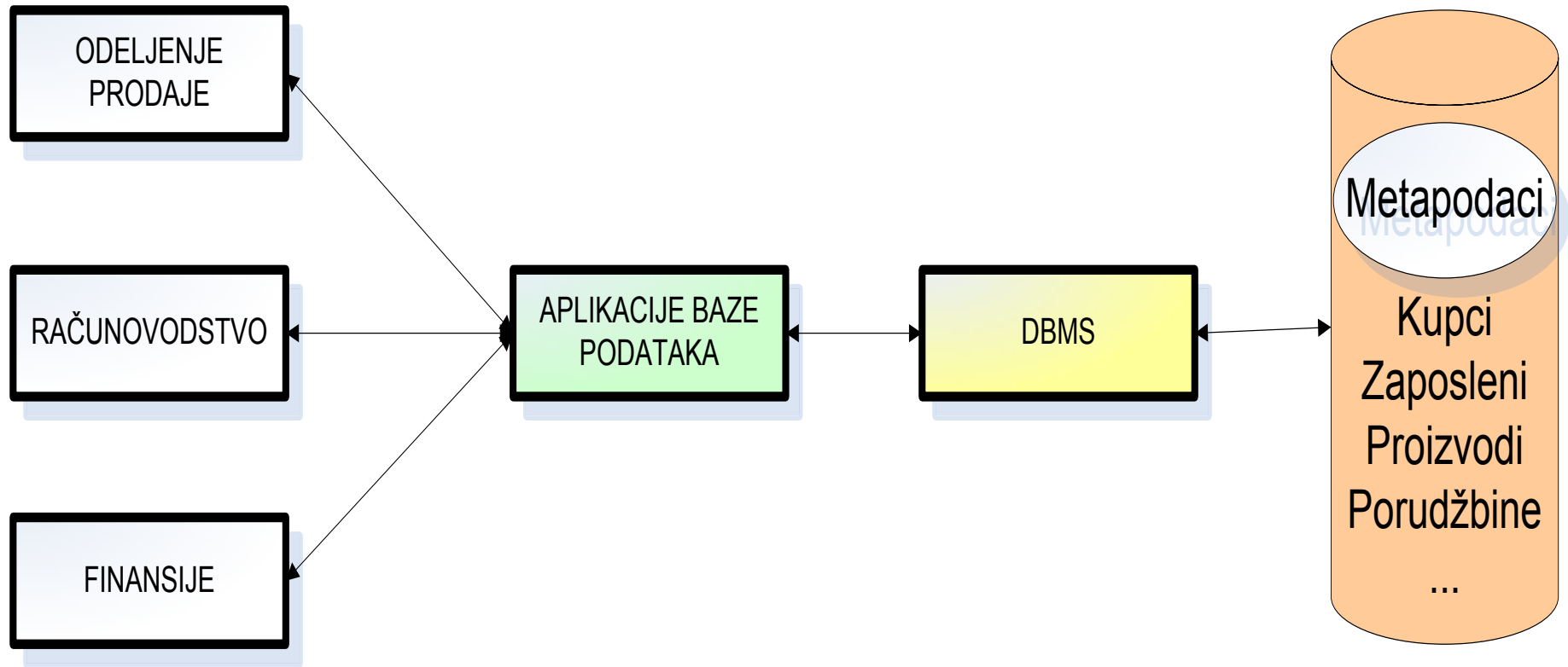
Klasični sistemi

- nedostaci -

- **Zavisnost između programa i podataka**
 - Svaki program mora da poznaje detaljan opis datoteka
- **Redudansa podataka**
 - Pojavljivanje istih podataka u različitim datotekama
- **Ograničenost deljenja podataka**
 - Zasebni podaci odeljenja, nema integrisanih izveštaja
- **Dugo vreme za razvoj**
 - Nema nastavka razvoja, najlakše je ispočetka
- **Teško održavanje programa**
 - 80% budžeta za razvoj se troši na održavanje

- Potencira **integraciju i deljenje podataka između svih odeljenja jedne organizacije**
- Zahteva potpunu promenu u načinu razmišljanja, na svim nivoima upravljanja
- Podaci koji su prethodno čuvani u više različitih datoteka, sada su **integrirani u jedinstvenu BP**
- Zajedno sa podacima čuvaju se i **metapodaci**
- Postoji posebna softverska komponenta – **DBMS (SUBP)** – interfejs između programa i podataka

Pristup BP



Pristup BP

- prednosti -

1. Nezavisnost između programa i podataka

- Odvajanje metapodataka od aplikacija koje koriste podatke
- Omogućen prenos podataka organizacije na druge računarske sisteme bez potrebe za promenom programa

2. Minimalna redudansa podataka

- **Podaci su integrisani** u jedinstvenu logičku celinu
- Svaki **podatak se nalazi samo na jednom mestu u BP**

3. Poboljšana konzistentnost podataka

- Ne postoji redudansa podataka, i **smanjene su greške**

Pristup BP

- prednosti -

4. Poboljšana razmena podataka

- BP je resurs cele organizacije
- Korisnici imaju različite poglede na jedinstvenu BP

5. Povećana produktivnost u razvoju aplikacija

- Smanjeni su troškovi za razvoj novih aplikacija
- Programeri razmišljaju o funkcijama, a ne o detaljima opisa podataka ili implementaciji

6. Smanjena potreba za održavanjem programa

- Moguće je nezavisno promeniti format podataka ili aplikaciju

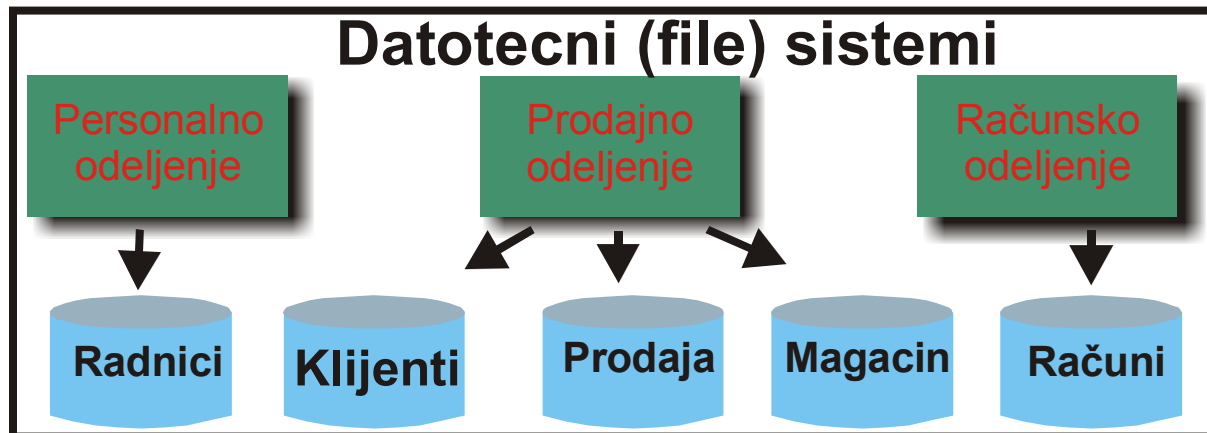
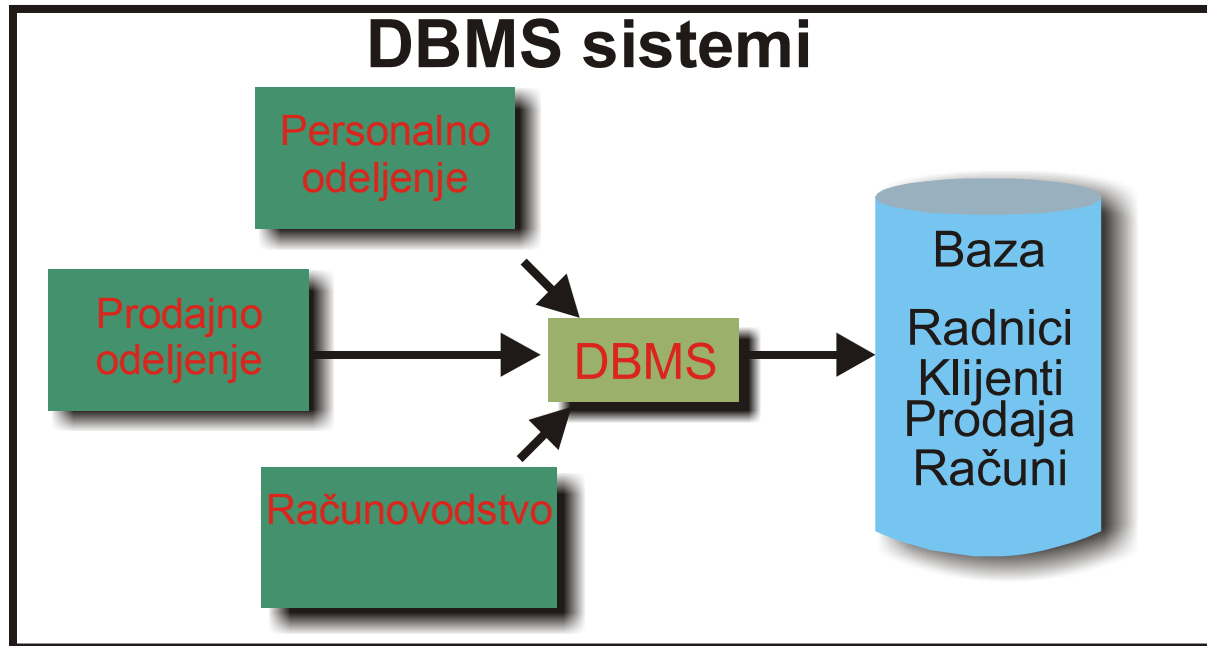
Pristup BP

- rizici -

1. Novo, obučeno osoblje
 - **Promene tehnologije, neprekidne obuke**
2. Troškovi i složenost instaliranja, upravljanja i rada sistema sa bazama podataka
 - **Dodatni hardver i softver**
3. Troškovi konvertovanja podataka
4. Potreba za izradom **sigurnosnih kopija i oporavkom podataka (*backup*)**
5. Konflikti u organizaciji

DBMS

Razlika DBMS i fajl sistema



DBMS (SUBP)

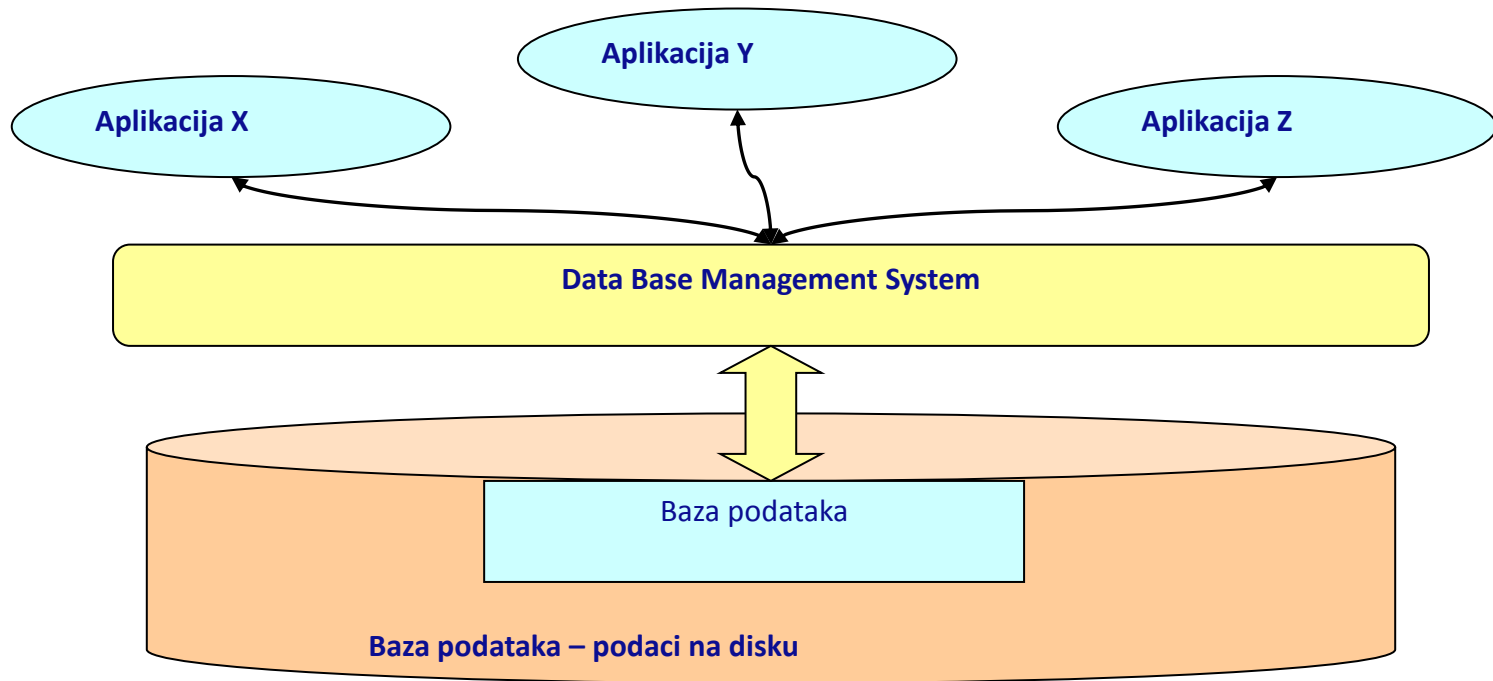
DBMS - *Data Base Management System*

- Sistem za upravljanje bazama podataka
- Softverski sistem koji se koristi za:
 - kreiranje,
 - održavanje i manipulisanje podacima,
 - kontrolu prava pristupa bazi podataka

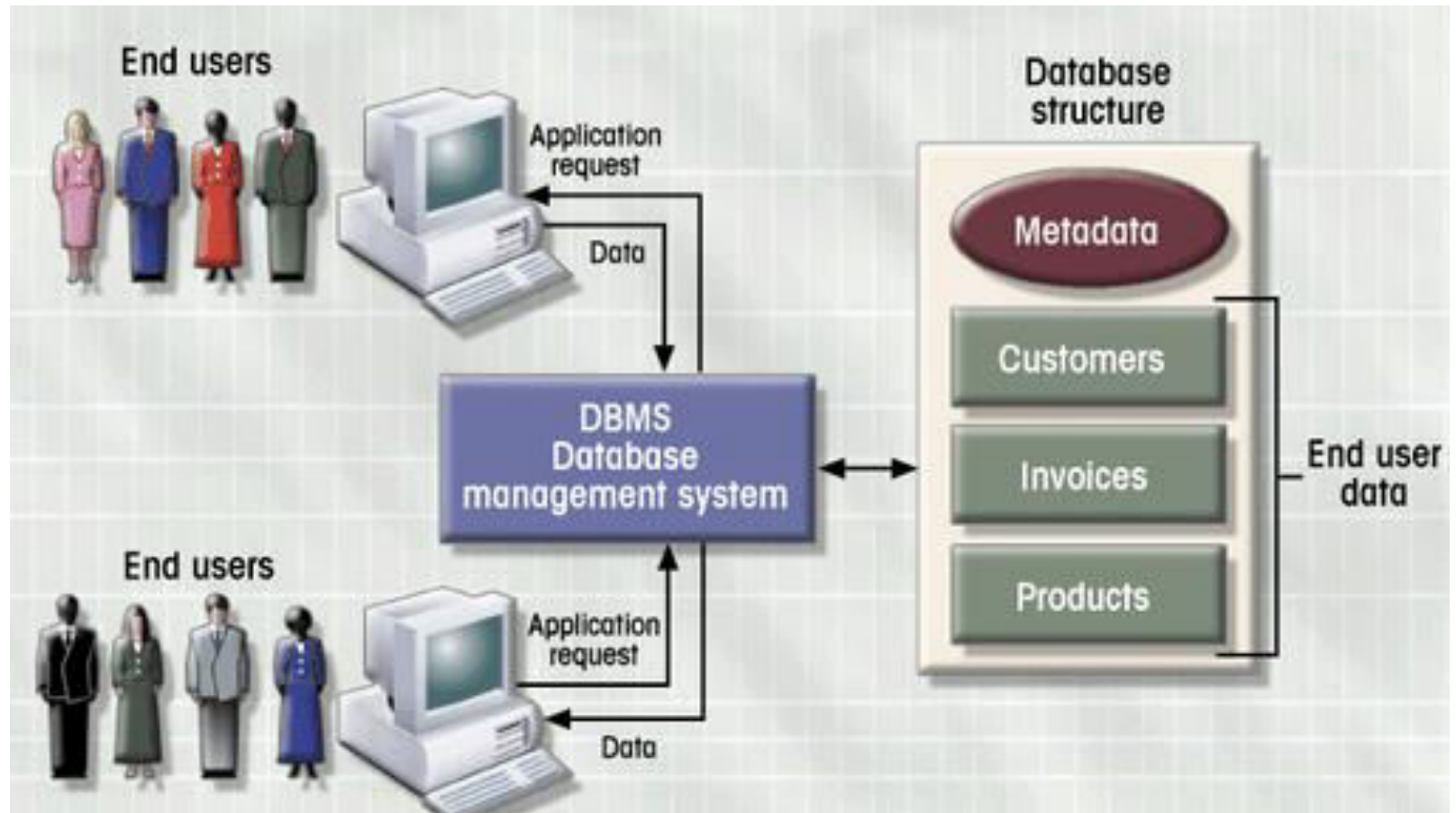
Database management systems - DBMS

- **Sistem za upravljanje bazom podataka** ili **SUBP** (*database management system, DBMS*) je skup programa, koji su neophodni za definisanje, kreiranje, rukovanje, upravljanje i korišćenje baza podataka.
- Primeri DBMS softvera:
 - *Oracle 12c, Oracle Corporation*
 - *IBM DB2*
 - *Microsoft SQL Server*
 - *SAP Sybase ASE*
 - *MySQL*
 - *Access*
- Ovi sistemi modeliraju podatke realnog sistema, čuvaju ih nezavisno od aplikacija i olakšavaju korisnicima pristup, dodavanje, sortiranje, pronalaženje i štampanje podataka. DBMS sistemi poseduju sopstveni upitni jezik koji omogućava brzo pronalaženje bitnih informacija.

- **DBMS** - Interfejs između korisnika (korisničkih programa, aplikacija) i zapisa baze podataka na disku



Sistem za upravljanje bazom podataka (SUBP)



Zadaci SUBP-a

- Omogućava efikasno upravljanje podacima, unos podataka, brisanje i modifikacija sadržaja
- Obezbeđuje jezik upita koji omogućava davanje odgovora na upite
- Obezbeđuje bolji pristup velikoj količini podataka kao i bolju organizaciju podataka
- Smanjuje verovatnoću nekonzistentnosti podataka
- Zaštita i sprečavanje neovlašćenog pristupa
- Prezentacija i vizuelizacija podataka

Karakteristike sistema za upravljanje bazama podataka

- Nezavisnost programa od podataka
- Kontrolisana redundansa
- Integritet podataka
- Sigurnost podataka
- Pristup zajedničkim podacima
- Konkurentni pristup
- Oporavak baze podataka



Administrator baze podataka je osoba koja se bavi organizacijom, kontrolom i zaštitom podataka u bazi.

DBMS (SUBP)

- **Nezavisnost programa od podataka** se odnosi na razdvajanje logičke i fizičke strukture podataka, tako da izmena fizičke strukture ne utiče na logičku strukturu podataka. Time se postiže nezavisnost aplikativnih programa na izmene u fizičkoj organizaciji podataka ili u logičkoj organizaciji usled dodavanja ili izmene strukture podataka.

DBMS (SUBP)

- **Kontrolisana redundansa** podataka podrazumeva izbegavanje višestrukog pamćenja istih podataka. **Redundancija** (ponavljanje) podataka se ne može sasvim izbeći, ali se mora svesti na minimum.
- **Osnovno pravilo dobro projektovanih baza podataka je pamćenje jednog podatka samo na jednom mestu.**
- Redundancija podataka je **obično** karakteristika loše dizajniranih sistema.

DBMS (SUBP)

- **Integritet podataka** podrazumeva tačnost, ispravnost podataka u bazi.
- Integritet podataka u bazi može biti narušen usled različitih razloga: namernog ili slučajnog unosa pogrešnih podataka, hardverskih neispravnosti i greškama u projektovanju baze podataka. Problem integriteta podataka je veći u sistemima kojima pristupa više korisnika.
- Da bi se izbegle greške u integritetu podataka, programi za rad sa bazama podataka bi trebali da koriste procedure za validaciju (proveru ispravnosti) podataka pri unosu i ažuriranju podataka, procedure za proveru prava pristupa i fizičko obezbeđenje.

DBMS (SUBP)

- **Sigurnost podataka** je zaštita podataka od neovlašćenog pristupa u bazi podataka.
- Sigurnost podataka je naročito bitna za tajne podatke kao što su razni lični podaci, zdravstveni, finansijski itd. Podjednako važna je i zaštita od gubitka podataka usled kvara računarske opreme.

DBMS (SUBP)

- **Konkurentni pristup** podacima označava istovremeni pristupanje jednom podatku od strane više korisnika.
- Ovakvi zahtevi su česti u sistemima sa velikim brojem korisnika i mogu da stvore različite konfliktne situacije koje ugrožavaju integritet baze podataka. Jedan od primera konflikta je pokušaj istovremenog ažuriranja nekog podataka, pri čemu se ažuriranja podatka od strane jednog korisnika poništava akcijom ažuriranja drugog korisnika.
- Za prevenciju konflikata koji se javljaju pri konkurentnom pristupu podacima koristi se princip *zaključavanja* slogova kojim se sprečava pristup ostalim korisnicima sve dok se slog ne otključa, odnosno oslobodi za druge korisnike, što se najčešće dešava nakon editovanja sloga.

DBMS (SUBP)

- **Oporavak baze podataka** (data recovery) je neophodan postupak u slučaju narušavanja integriteta podataka iz bilo kog razloga.
- Da bi se mogao sprovesti postupak oporavka baze podataka i u cilju zaštite integriteta podataka vrši se plansko pravljenje rezervnih kopija podataka (back up). (dnevno, sedmično, mesečno, godišnje)
- Ovaj postupak se naziva *restauracija* (restore) oštećene baze podataka. Nakon procesa restauracije baze podataka, ponovo se uspostavlja integritet podataka, ali se gube podaci koji su dobijeni u obradama nakon poslednjeg back up-a.

DBMS (SUBP)

- Uglavnom je teško i nesigurno ponoviti sve transakcije, jer se korisnici ne mogu setiti koje izmene su vršili pre, a koje posle arhiviranja, kao i njihov redosled.
- Taj problem se rešava pravljenjem kopije ažuriranih slogova, pre (**before image**) i posle (**after image**) svake transakcije.
- Before image kopije se koriste za poništavanje ažuriranja, tako što se kopiraju ponovo u bazu u suprotnom redosledu od redosleda nastajanja (roll back).
- After image kopije se koriste za ažuriranje slogova tako što se kopiraju ponovo u bazu onim redosledom kojim su i nastale (roll forward).

- **Registar korisnika**

- Podaci o korisnicima i njihovim pravima pristupa
- Ko može da pristupa
 - Username, password
- Kojim podacima
 - Baza podataka, tabele, kolone, vrste
- Šta može da radi sa njima
 - INSERT, DELETE, UPDATE, SELECT

RELACIONE BP

ŠTA JE ZA NAS ZNAČAJNO ?

- Za nas su značajne:
 - Relacione baze podataka - RBP
 - Programi za upravljanje sa RBP – Sistemi za upravljanje bazama podataka (DBMS)

Relational Database Management System (RDBMS)
- Reč “relaciona” danas se često izostavlja, zato što većina poznatih SUBP (kao što su: MS Access, MySQL, MS SQL Server, Oracle, IBM DB2, Informix, Sybase) jesu relacione BP.
- Relacioni model – opisuje određeni aspekt stvarnog sveta (*problem space*) u skladu sa pravilima (E.F.Codd krajem 60. godina XX veka)

- **Karakteristike relacionih BP**

- Podaci su organizovani u redove i kolone, a skup takvih podataka je relacija
- Sve vrednosti su skalarne – u jednom polju se može naći samo jedna vrednost
- Ne mogu da postoje dupli zapisi
- Sve operacije se obavljaju nad relacijama a rezultat operacija je opet relacija

- **Koncept RBP:**

- Na slici su prikazane 3 tabele sa ukupno 12 atributa (različitih kolona)
- Tab1 i Tab2 “dele” atribut 3
- Tab2 i Tab3 “dele” atribut 7

Tab1

Attrib. 1	Attrib. 2	Attrib. 3	Attrib. 4	Attrib. 5

Tab2

Attrib. 3	Attrib. 6	Attrib. 7

Tab3

Attrib. 8	Attrib. 9	Attrib. 10	Attrib. 11	Attrib. 7	Attrib. 12

- Postojanje relacije potpuno je nezavisno od fizičkog oblika podataka
 - Relacija može da bude fizička tabela na disku
 - Relacija može da bude i izvedena iz više table, pojedine vrednosti mogu biti dodati (izračunati) itf.

Primena BP predstavlja kombinovanu upotrebu:

- Relacionog sistema za upravljanje bazama podataka (DBMS)
- Relacione BP
- Odgovarajućeg menija, formi za unos podataka i generisanih izveštaja
- Programa za obradu podataka
- Dokumentacije za korisnika (npr. *manuals*)

Osnovne komponente BP

- Baza podataka uključuje u sebe sledeće komponente:
 - uskladišteni podaci
 - informacioni jezici za opis i manipulaciju podacima
 - sistem za upravljanje bazom podataka (SUBP)
 - alati za administraciju baze podataka
 - rečnik baze podataka (metapodaci)
 - tehnička sredstva baze podataka

**ENTITET, INSTANCE,
ATRIBUTI I IDENTIFIKATORI**

Entiteti

- Ako znate kako da organizujete i klasifikujete podatke znaćete da izvućete korisne zaključke o podacima
- Važno je naučiti šta su **entiteti** jer su to stvari o kojima čuvamo podatke.
- **Entitet je bilo šta o čemu je moguće prikupljati informacije (ljudi, događaji, objekti)**

Na primer:

Školi je potrebno da smeste podatke o (minimalno):
STUDENTIMA, NASTAVNICIMA, PREDMETIMA,
UČIONICAMA, OCENAMA

Entitet je:

- “Nešto” od važnosti za posao o čemu moraju da se znaju podaci
- Ime za grupu sličnih stvari koje možete da navedete
- Obično imenica u jednini
- Primeri: stvari (auto, knjiga), događaji (utakmica, ispit), ljudi (student, radnik)

Atributi

- Važno je naučiti o **atributima** jer oni **pružaju više informacija o entitetima**.
- **Atributi** pomažu da budemo izričitiji (precizniji) o tome koje podatke želimo da pratimo.

Na primer:

U restoranu, treba vam spisak različitih stavki narudžbine da biste znali koliko da naplatite.

Pri pravljenju više izveštaja o prodaji za jedno odeljenje, morate odabrati prave izveštaje prodaje iz liste izveštaja.

Šta je **atribut**?

Kao i **entitet**, i **atribut** predstavlja nešto od važnosti za posao.

Atribut je poseban deo informacije koji:

- Opisuje entitet
 - Kvantifikuje entitet
 - Kvalifikuje entitet
 - Klasifikuje entitet
 - Specifikuje entitet
- ✓ Atribut ima samo jednu vrednost za jednu instancu entiteta.

Još o atributima

- Neki atributi moraju da imaju vrednost. To su **MANDATORNI** atributi. Na primer: u većini poslova koji prate informacije o zaposlenima, *ime* je potrebno.
- Drugi atributi mogu imati vrednost ili ostati nepopunjeni. To su **OPCIONALNI** atributi. Na primer: *broj telefona* često nije potreban, izuzev u mobilnim ili bežičnim aplikacijama.
- *E-mail adresa* može biti mandatorni atribut za ZAPOSLENOG ako modelujete e-mail aplikaciju. Može biti opcionalan za MUŠTERIJU ako modelujete online katalog.
- Ili, svaka osoba mora da ima *datum rođenja*, a može da ima *datum venčanja*, *datum diplomiranja* i sl.

ENTITETI	Atributi
MUŠTERIJA	Prezime, godine starosti, broj cipela, mesto stanovanja, e-mail
KOLA	Model, težina, katalogska cena
NARUDŽBINA	Datum naručivanja, datum isporuke
POSLO	Naziv, opis
TRANSAKCIJA	Iznos, datum transakcije
RADNI UGOVOR	Datum početka, plata

MUŠTERIJA	Prezime, starost, broj cipela, grad stanovanja, e-mail
KOLA	Model, težina, katalogska cena
NARUDŽ BINA	Datum naručivanja, datum isporuke
POSLO	Naziv, opis
TRANSAKCIJA	Iznos, datum transakcije
RADNI UGOVOR	Datum početka, plata

- Atributi imaju vrednosti.
- **Vrednost** atributa može biti broj, niz znakova, datum, slika, zvuk, itd.
- To su takozvani *tipovi podatka* ili *formati*. Svaki atribut ima tip podatka.

Entiteti

opisuju

Atributi

koji pripadaju
određenom

Tipu
podataka

Instance

su opisane

Vrednostima atributa

koje se
pamte kao

Podaci

Struktura baze podataka
(konceptualni model)

Entitet - Instanca

Atributi -

Vrednosti atributa

Sadržaj baze podataka
(fizička implementacija)

Instanca

Entiteti imaju **instance**.

Instanca entiteta je jedinstvena **pojava entiteta**.

Entitet

Instanca

OSOBA

Mahatma Gandhi

PROIZVOD

Cipele Nike

TIP PROIZVODA

Cipele

POSLO

Električar

NIVO SPOSOBNOSTI

Početak

KONCERT

U2 na Paladijumu

ŽIVOTINJA

Pas

KOLA

Volkswagen buba

Entiteti i instance

Dalmatinac, Sijamska mačka, krava i prase su instance entiteta ŽIVOTINJA

Kabriolet, sedan, prikolica su instance entiteta VOZILO

Postoji mnogo entiteta.

Neki entiteti imaju mnogo instanci; neki imaju samo poneku.

Entiteti mogu biti:

- Opipljivi, kao OSOBA ili PROIZVOD
- Neopipljivi, kao NIVO SPOSObNOSTI
- Događaj, kao KONCERT

Mogu nam biti važni entiteti i instance ako:

- postoje,
- postojali su, ili
- će postojati

Na primer:

- Važno nam je da znamo koliko robe postoji na skladištu da bismo znali koliki prihod možemo da ostvarimo njenom prodajom.
- Odigrane utakmice pamtimo da bismo na kraju prvenstva utvrdili prvenstvenu tabelu.
- Za stambenu zgradu koja je projektovana ali još nije izgrađena vlasnik može da izračuna koliki prihod će imati od stanarina kada bude završena i izdata.

Da li je PAS instanca ili entitet?

ZAVISI

Šta ako smo zainteresovani za različite vrste životinja? Tada ima smisla razmišljati o **entitetu ŽIVOTINJA** sa instancama **PAS, MAČKA, KONJ** i tako dalje.

Šta ako vodimo uzgajivačnicu pasa? Moramo da čuvamo podatke o različitim rasama pasa, ali ne o nekim drugim vrstama životinja. Za uzgajivačnicu pasa, mnogo je prirodnije misliti o **entitetu PAS** sa instancama **TERIJER, PUDLA, LABRADOR** i tako dalje.

Jedinstven identifikator

(UID – unique identifier)

Pogledajte reklame u magazinima i internet sajtove.

Koja je “glavna stvar” kod svake reklame ili veb sajta?



carmax.com

Jedinstven identifikator (UID)

- Razmislite o svim studentima u učionici.
- Svaki student je opisan u nekoliko crta ili atributa.
- Koji atribut ili atributi vam dozvoljavaju da odaberete jednog studenta od ostatka grupe?

To je jedinstveni identifikator studenta.

Jedinstven identifikator (UID)

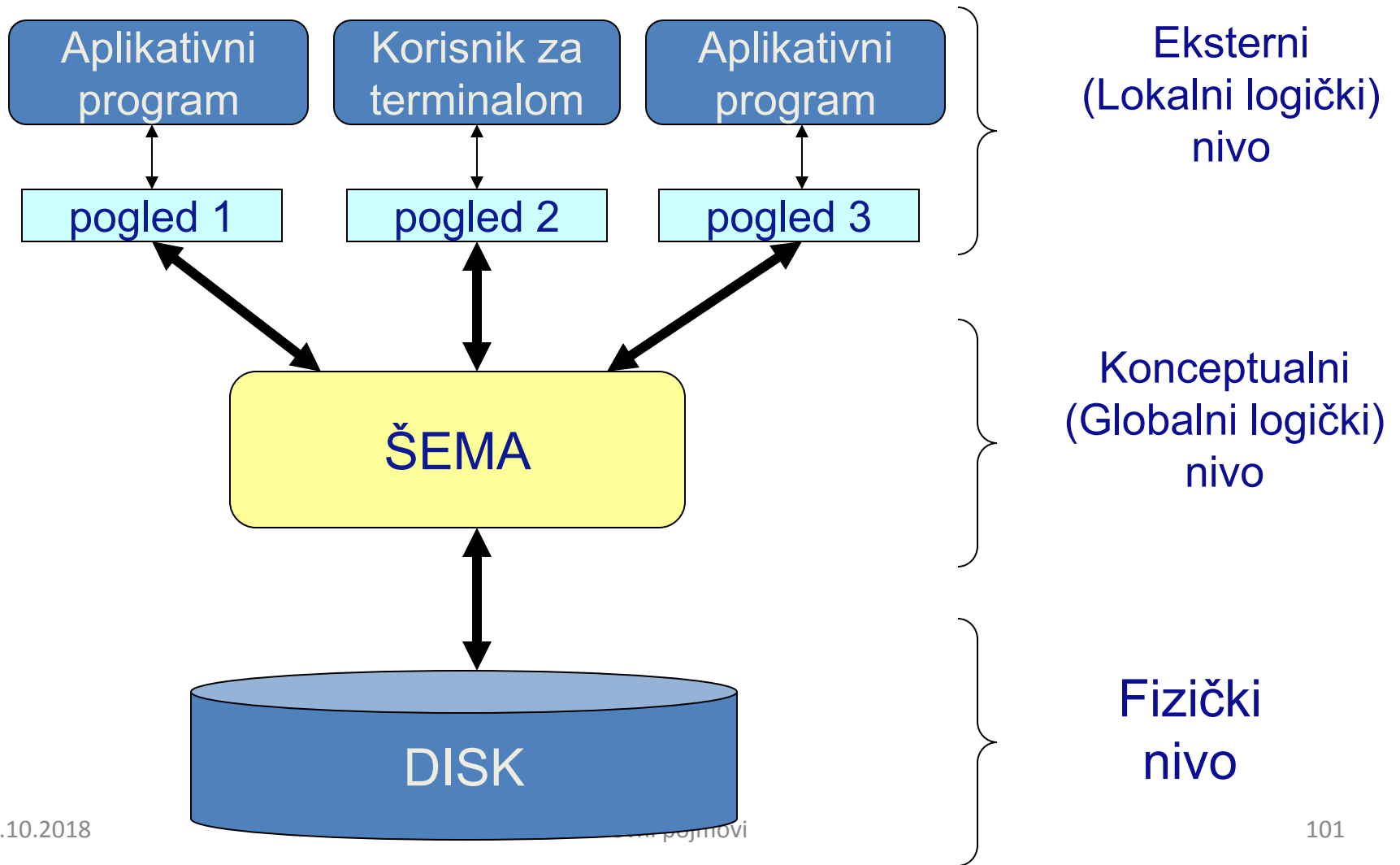
- Potreban vam je jedinstven identifikator (UID) za pesme u vašoj CD-kolekciji.
- UID je atribut ili kombinacija atributa koji razlikuju jednu pesmu od druge.
- Kako da nađete određenu pesmu u celoj kolekciji?
- Koji bi bio jedinstveni identifikator za PESMU?

MODELOVANJE BP

Standardizacija

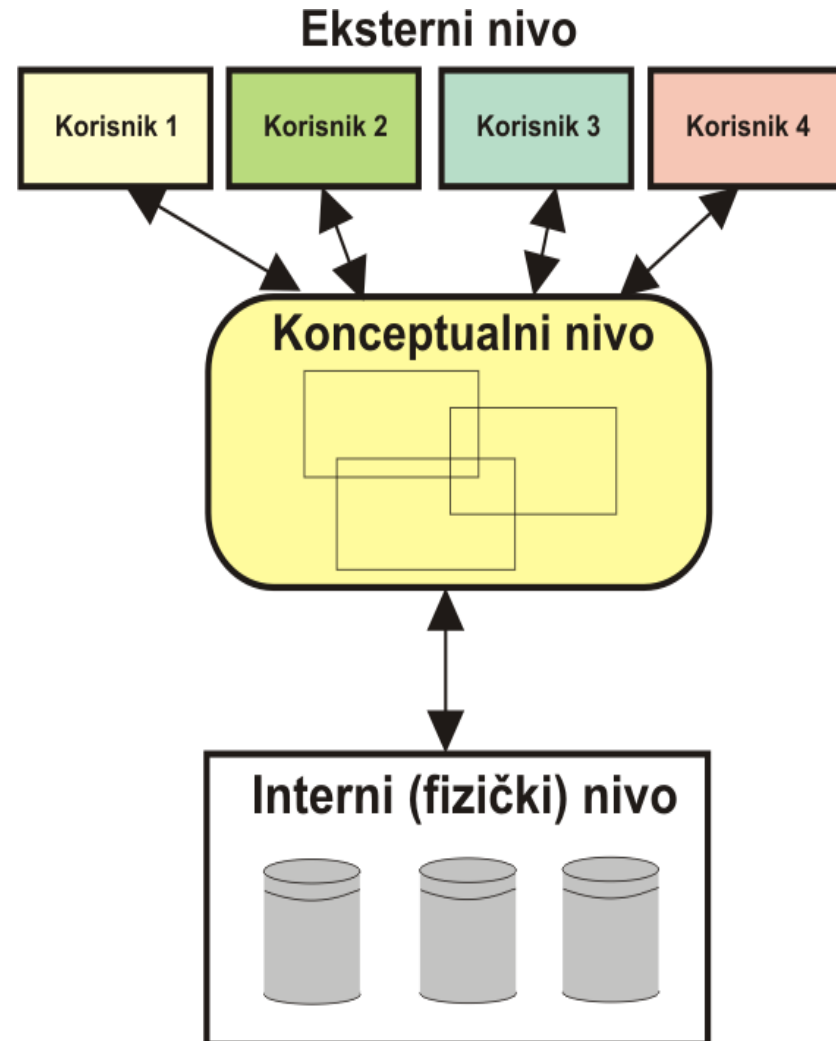
- Slojevi (*layers*) idealne BP
 - Slojevit rad – uvođenje standarda
 - Podela na 3 nivoa
 - Eksterni (Korisnik) View[1], ..., View[n]
 - Konceptualni Opis preduzeća
 - Interni Datoteke, metod pristupa, indeksi
 - Različiti slojevi imaju različite zadatke
 - Nezavisnost slojeva

Arhitektura BP



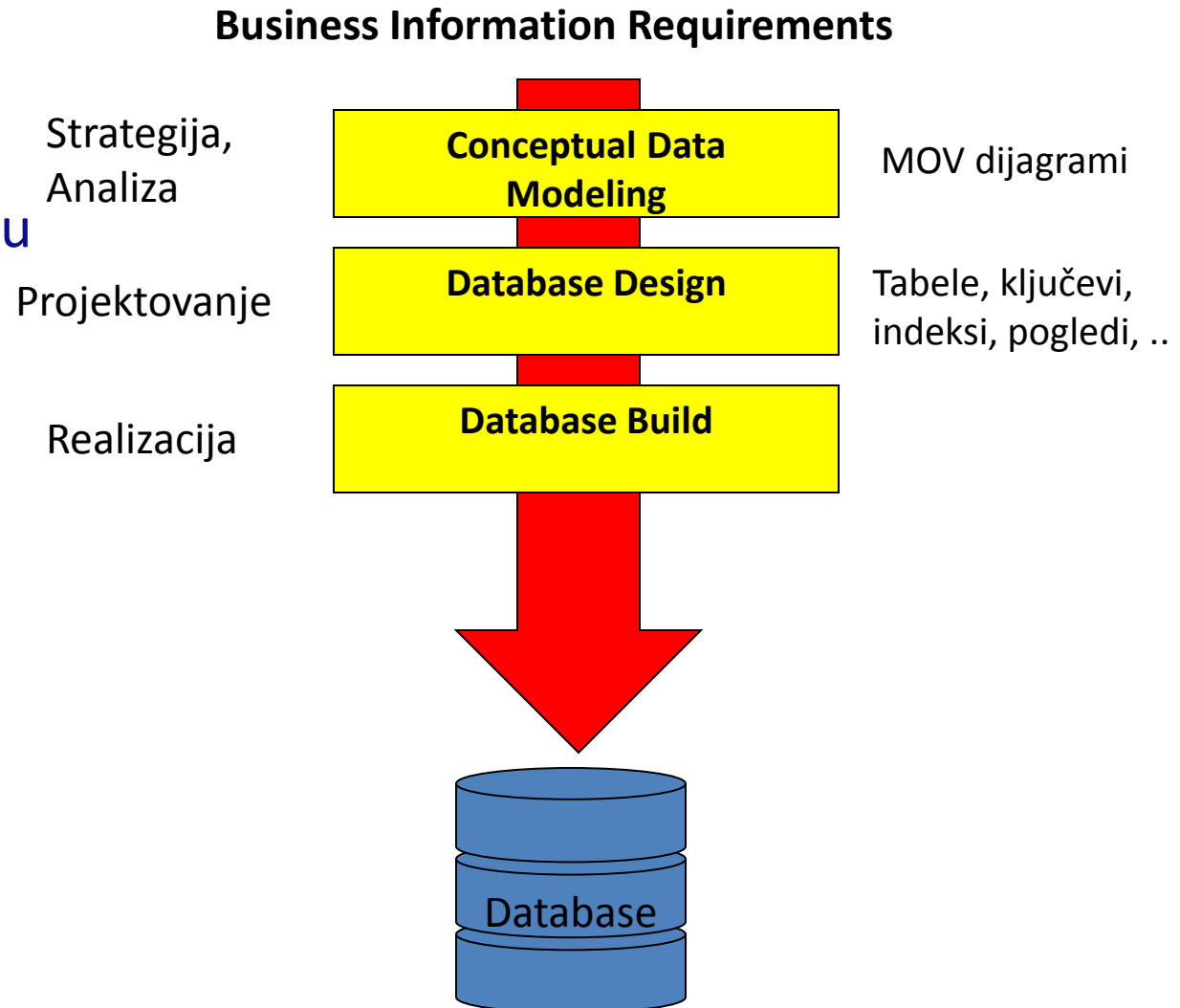
DBMS arhitektura na tri nivoa

- **Konceptualni nivo** prikazuje logički model celokupne baze podataka kojim je definisan sadržaj baze podataka. Logički nivo je predstavljen šemom baze podataka, u okviru kojih su definisani atributi za svaku vrstu entiteta koji je obuhvaćen u modelu.
- **Eksterni (korisnički) nivo** se sastoji od skupa korisničkih modela. Korisnički model predstavlja pogled na jedan deo baze podataka koji je od interesa određenom korisniku. Definisanje korisničkih nivoa je neophodno radi određivanja prava pristupa delovima baze podataka. Eksterni nivo je predstavljen podšemom baze podataka.
- **Interni (fizički) nivo** obuhvata sadržaj cele baze podataka i definiše način fizičke organizacije podataka na medijumima gde su podaci smešteni (formati zapisa u memoriji, organizacija i drugo.).



Modelovanje - Razvoj BP-

- Modelovanje podataka je prvi korak u razvoju baza podataka (*database development process*)



Modelovanje

- Razvoj BP-

- Razvoj baza podataka počinje sa analizom zahteva krajnjih korisnika
- Primer - Problem upravljanja kadrovima (zaposleni):
 - Potrebno je voditi evidenciju za svakog zaposlenog (ime, prezime, radno mesto, datum zapošljavanja, zarada)
 - Svaki zaposleni ima jedinstven identifikacioni broj
 - Preduzeće ima više odeljenja (proizvodnja, računovodstvo, prodaja itd). Odeljenja imaju jedinstven ID
 - Svaki zaposleni je u jednom odeljenju
 - Neki od zaposlenih su direktori odeljenja
 - Potrebno je da se zna direktor za svakog zaposlenog i svi zaposleni kojima upravlja određeni direktor

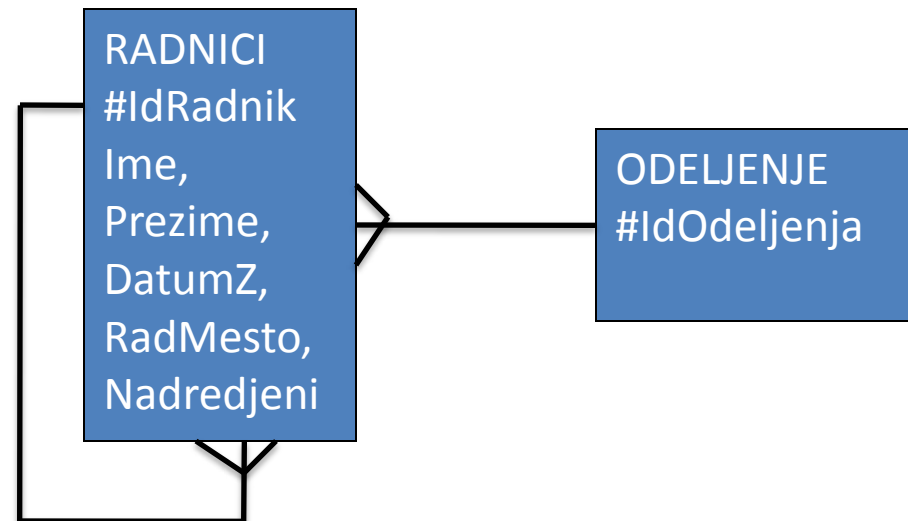
KONCEPTUALNI MODEL (I)

Modelovanje

- Razvoj BP-

Primer ER modelovanja

- ER model pokazuje odnose u datoj organizaciji i podržava poslovne funkcije



Modelovanje

- Razvoj BP-

- *Database design*: ER model se transformiše u Relacioni model:
 - Nazivi tabela
 - Nazivi kolona u tabelama
 - Primarni (PK) i spoljašnji ključevi (FK)
 - Nulls: indikacija da kolona mora da ima vrednost
 - Unique: vrednost u datoj koloni ne sme da se ponavlja
 - Datatype: format i definicija svake kolone

PROJEKTOVANJE BP (II)

Zašto učiti projektovanje baza podataka?

DA BI RAZLIKOVALI
DOBAR I ~~LOŠ~~ DIZAJN BAZA PODATAKA



Zašto je bitan dobar dizajn baze podataka ?

- Dizajn baze podataka je *delikatna* stvar i treba joj pristupiti *oprezno*, jer početne greške u dizajnu dovode do velikih problema u daljem korišćenju baze podataka
- ZDRAVORAZUMSKI DIZAJN = može dovesti do toga da nova baza podataka u potpunosti odslikava postojeći “paper – based system”

- Naizgled dobro formirana NARUDŽBENICA

<u>Sifra kupca</u>	Ime	Lokacija	Telefon	<u>Šifra artikla</u>	Naziv	Cena	Količina
--------------------	-----	----------	---------	----------------------	-------	------	----------

■ **Ipak loša**

<u>Sifra kupca</u>	Ime	Lokacija	Telefon	<u>Šifra artikla</u>	Naziv	Cena	Količina
K1	LEONA	Novi Sad	024-455-44	A1	Penkala	120 din	200
K1	LEONA	Novi Sad	024-455-44	A2	Spajalice	50 din	1000
K2	ZORA	Čačak	032-345-67	A1	Penkala	120 din	85
K3	GALEB	Beograd	011-234-345	A3	Mastilo	350 din	800
...	...	...	...	...	...	...	...
K1	LEONA	Novi Sad	024-455-44	A3	Mastilo	350 din	20

Projektovanje BP

- Neophodnost dobrog projektovanja BP:
 - Važno je pažljivo razmišljati o tabelama i njihovoj strukturi
 - Poželjno je da se podaci pojavljuju samo jedanput
 - U datom primeru se **DatumRođenja** pojavljuje dva puta
 - Opasnost kod višestrukog pojavljivanja **Adrese** zaposlenih
 - Ako se zaposleni preseli, promena se mora evidentirati na više mesta
 - Izuzetno, zbog efikasnosti – brzog pristupa podacima, nekada je potrebno imati redudansu

Zašto je bitan dobar dizajn baze podataka ?

- Najveći problem kod loše dizajniranih baza podataka nastaje usled postojanja **redundantnih podataka**, što označava nepotrebno dupliranje istih podataka u bazi. Pojava postojanja duplih podataka naziva se **redundancija**.
- Primer redundancije, je recimo kada se broj telefona kupca pojavljuje u datoteci o kupcima, u datoteci računa i datoteci agenata prodaje u okviru iste baze podataka.

Nekontrolisana redundancija podataka je tipična za loše dizajnirane baze podataka

Problemi redundancije

- **Nekonzistentnost podataka** se javlja kada se isti podatak pojavljuje na različitim mestima sa vrednostima koje su međusobno konfliktne.
- Izveštaji koji se prave na osnovu ovih podataka daju različite vrednosti, zavisno od upotrebljene verzije podatka. Tada se kaže da je u toj bazi podataka narušen **integritet podataka**, jer se dobijaju različiti rezultati za isti podatak.

Redundancija $\Rightarrow$ Anomalije podataka

1. *Anomalija ažuriranja.*
2. *Anomalija dodavanja.*
3. *Anomalija brisanja*

najviše

<u>Sifra kupca</u>	Ime	Lokacija	Telefon	<u>Šifra artikla</u>	Naziv	Cena	Količina
K1	LEONA	Novi Sad	024-455-44	A1	Penkala	120 din	200
K1	LEONA	Novi Sad	024-455-44	A2	Spajalice	50 din	1000
K2	ZORA	Đađak	032-345-67	A1	Penkala	120 din	85
K3	GALEB	Beograd	011-234-345	A3	Mastilo	350 din	800
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
K1	LEONA	Novi Sad	024-455-44	A3	Mastilo	350 din	20

Dobro dizajnirana baza podataka postaje generator vrednih informacija.

Loše dizajnirana baza podataka generiše pogrešne informacije, koje dalje vode pogrešnim odlukama u sistemima.

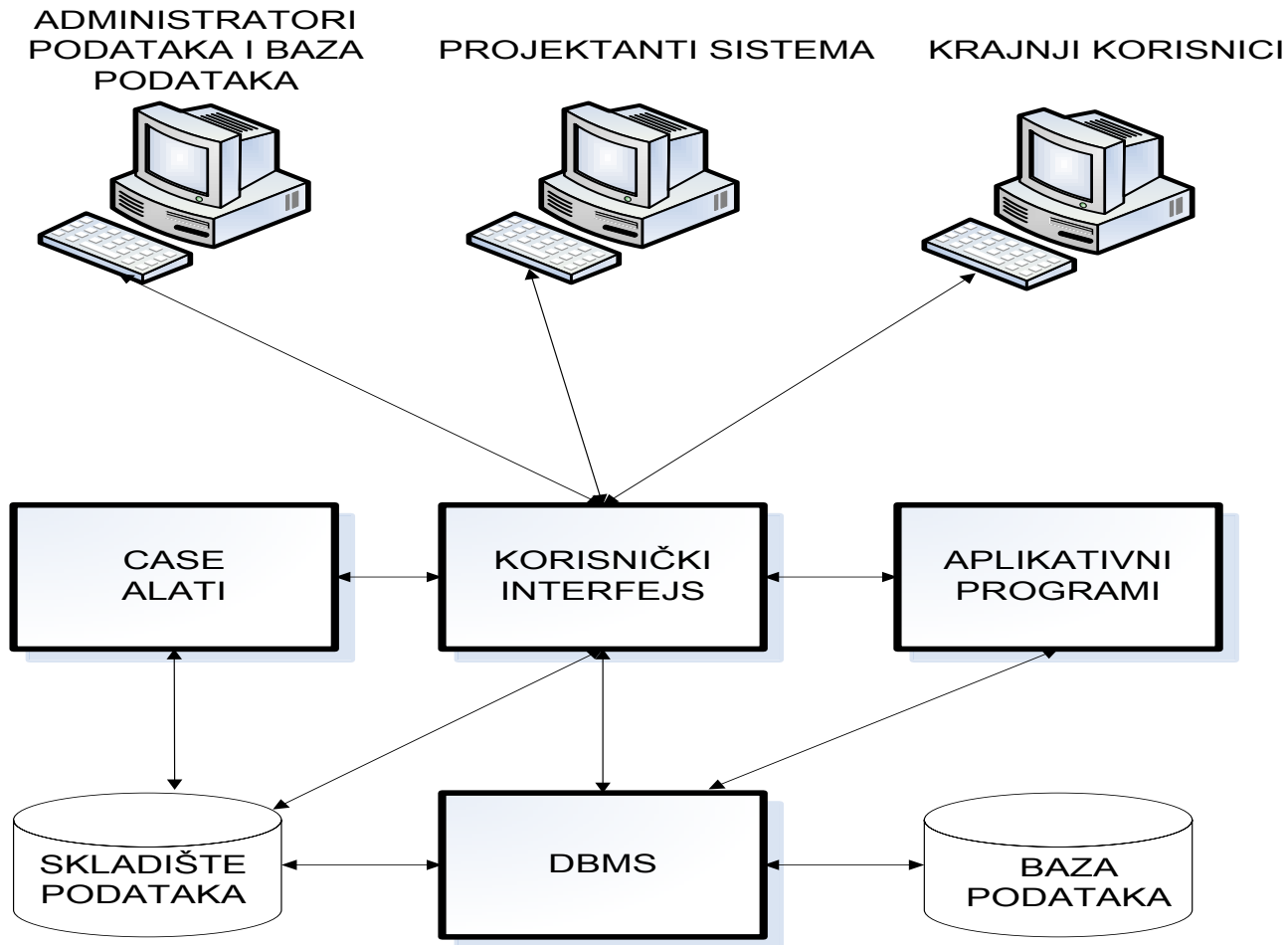
REALIZACIJA BP (III)

- Realizacija (*database build*)
 - Primena SQL (*Structured Query Language*) jezika za fizičku realizaciju

```
SQL>CREATE TABLE ODELJENJE  
      (IdOdeljenje NUMBER(5) NOT NULL PRIMARY KEY,  
       Naziv VARCHAR(25) NOT NULL);
```

```
SQL>CREATE TABLE RADNICI  
      (IdRadnik NUMBER(9) NOT NULL PRIMARY KEY,  
       Ime VARCHAR2(15) NOT NULL,  
       Prezime VARCHAR2(20) NOT NULL,  
       RadnoM VARCHAR2(15),  
       Zaposlen DATE NOT NULL,  
       Nadredjeni NUMBER(9) REFERENCES RADNICI  
       Odeljenje NUMBER(5) REFERENCES ODELJENJE);
```

Tipično okruženje BP



PITANJA ?