

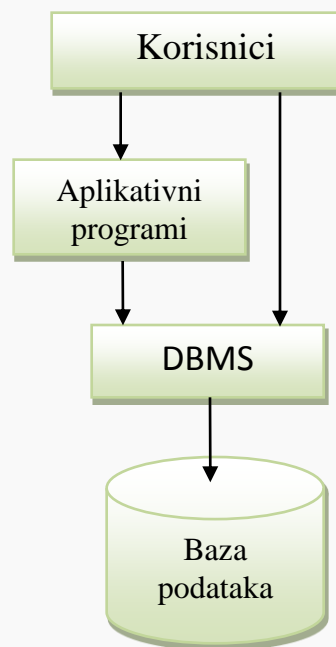
1 - Uvod

SADRŽAJ

- Produkcija informacija i potrebe za bazama podataka
- BP i Sistemi za Upravljanje Bazama Podataka
- Tipovi Baza podataka
- Pogled na podatke
- Modeli Podataka
- Šeme i instance
- Jezik za Definisanje Podataka (Data Definition Language)
- Jezik za Manipulaciju Podacima (Data Manipulation Language)
- Nedostaci file baziranih sistema sa BP i razlozi za uvođenje Sistema za Upravljanje Bazama Podataka
- DBMS: Ciljevi i osnovne komponente
- Klasifikacija DBMS
- Sveukupna Struktura relacionih DBMS
- Korisnici Baza Podataka
- Administrator Baze Podataka
- Aplikativna arhitektura

Produkcija informacija i baze podataka

- Obim produkcije informacija
- Uređenje kolekcije podataka u bazu
- Skladištenje podataka
- Informacije i podaci
- Komponente sistema:



Sistem sa bazom podataka

BP i Sistemi za Upravljanje Bazama Podataka (DBMS)

- Baza podataka: **kolekcija relevantnih podataka o sistemu od interesa i veza između njih.**
- Baza podataka sadrži informacije o određenom sistemu.
- DBMS: **Softverski sistemi opšte namjene za pogodan i efikasan pristup podacima u bazi podataka.**
- DBMS obezbjeđuje rješavanje zajedničkih aplikativnih problema i zadataka u radu sa bazom.
- Aplikacije koje rade sa bazama podataka:
 - ✎ Banke: sve transakcije u radu sa komitentima
 - ✎ Avio-prevoznici: rezervacije, rasporedi letova
 - ✎ Univerziteti: prijave, ocjenjivanje, ...
 - ✎ Prodaja: kupci, proizvodi, narudžbe, kupovine
 - ✎ Proizvodnja: izrada proizvoda, skladišta, narudžbe, nabavke
 - ✎ Ljudski resursi: lični podaci, plate, porezi i doprinosi
- Svi aspekti života pojedinaca i društvenih zajednica su povezani na određeni način sa različitim bazama podataka.

BP univerziteta sadrži podatke o:

STUDENT	ImeStudenta	JMB	GodinaStudija	StudijskiProgram
	Jovanović	12109 87 100018	3	RI
	Marković	05129 83 100067	4	RI
	Petrović	22089 88 105034	2	ET
	Živanović	06079 89 100021	2	ET

PREDMET	NazivPredmeta	IdPredmeta	ECTS
	Ekspertni sistemi	ES100	6
	Baze podataka	BP101	6
	Računarske mreže	RM102	6
	Internet Programiranje	IP100	7

POLAGANJE ISPITA	JMB	IdPredmeta	DatumIspita	Ocjena
	05129 83 100067	ES100	12/10/2009	7
	12109 87 100018	BP101	14/06/2011	9
	05129 83 100067	RM102	22/02/2010	9
	05129 83 100067	BP101	13/07/2010	8

Tipovi Baza Podataka

■ Hijerarhijske i mrežne organizacije podataka (period: 60 - 80-te godine).

- 👉 Zapisi organizovani u hijerarhijske ili mrežne strukture
- 👉 implementacija se bazirala na korištenju pokazivača.

■ Relacione baze podataka (period: 80 -).

- 👉 jednostavna logička organizacija podataka, logički u formi dvodimenzionalnih struktura – tabela, koje korespondiraju matematičkim relacijama

■ Objektno-orijentisane baze podataka.

- 👉 Bazu čini skup perzistentnih (trajnih) objekata koji sadrže kako podatke tako i metode za manipulaciju podacima i interakciju sa drugim objektima.
- 👉 Slični objekti pripadaju istoj klasi.
- 👉 Generalnost objektno-orijentisane paradigme i moćni koncepti (kompleksni tipovi podataka, enkapsulacija operacija i podataka, nasljeđivanje, jedinstveni identifikatori objekata ...)
- 👉 Kompleksnost objektno-orijentisanog modela i izostanak standardizacije u ranoj fazi razvoja koncepta ograničili primjenu objektno-orijentisanih BP
- 👉 Primjena je ograničena na specijalizovane aplikacije, kao na primjer u CAD (eng. *Computer Aided Design*) sistemima, proizvodnim sistemima itd.

Tipovi Baza Podataka

- **Djelomično strukturirane baze podataka** (eng. *Semistructured databases*),
 - 👉 omogućavaju postojanje podataka istih tipova koji mogu imati različit broj atributa.
 - 👉 XML (extensible markup language) - standardni jezik za specifikaciju djelomično strukturiranih podataka. Osnovna primjena za standardizovan, platformski nezavisan opis podataka i razmjenu informacija između heterogenih, računarski baziranih sistema. Slično OEM (Object Exchange Model) standard.
- **GIS baze podataka**
 - 👉 rade i efikasno manipulišu geografskim mapama i objektima, geološkim, infrastrukturnim i drugim prostornim podacima.
- **Data warehouse** baze podataka, generisane izdvajanjem sumarnih/agregiranih podataka iz velikih analitičkih baza
 - 👉 Koriste OLAP (eng. *online analytic processing*) tehnike, te moćne analitičke alate,
- **real-time** baze podataka,
 - 👉 koriste se u sistemima za upravljanje industrijskim, proizvodnim i drugim procesima kod kojih je bitan zahtjev za odziv u stvarnom vremenu,
- **big-data** baze podataka itd.

Nivoi Apstrakcije Podataka u Sistemima sa BP

PREDSTAVA PODATAKA

- **Fizički nivo** – opisuje kako su zapisi (npr. o studentima) fizički smješteni na odgovarajućem mediju.
- **Logički nivo: opisuje sveukupnu logičku strukturu baze podataka i veza između njih.**
 - 👉 Opis može da bude na **konceptualnom nivou**,
 - 👉 ili **prezentacionom (implementacionom) nivou**. Opisuje koji podaci (određenih tipova) se smještaju u bazu podataka, i kakve su relacije između podataka.

type student = record

ImeStudenta: string;

JMB: integer;

GodinaStudija: integer;

StudijskiProgram: string;

end;

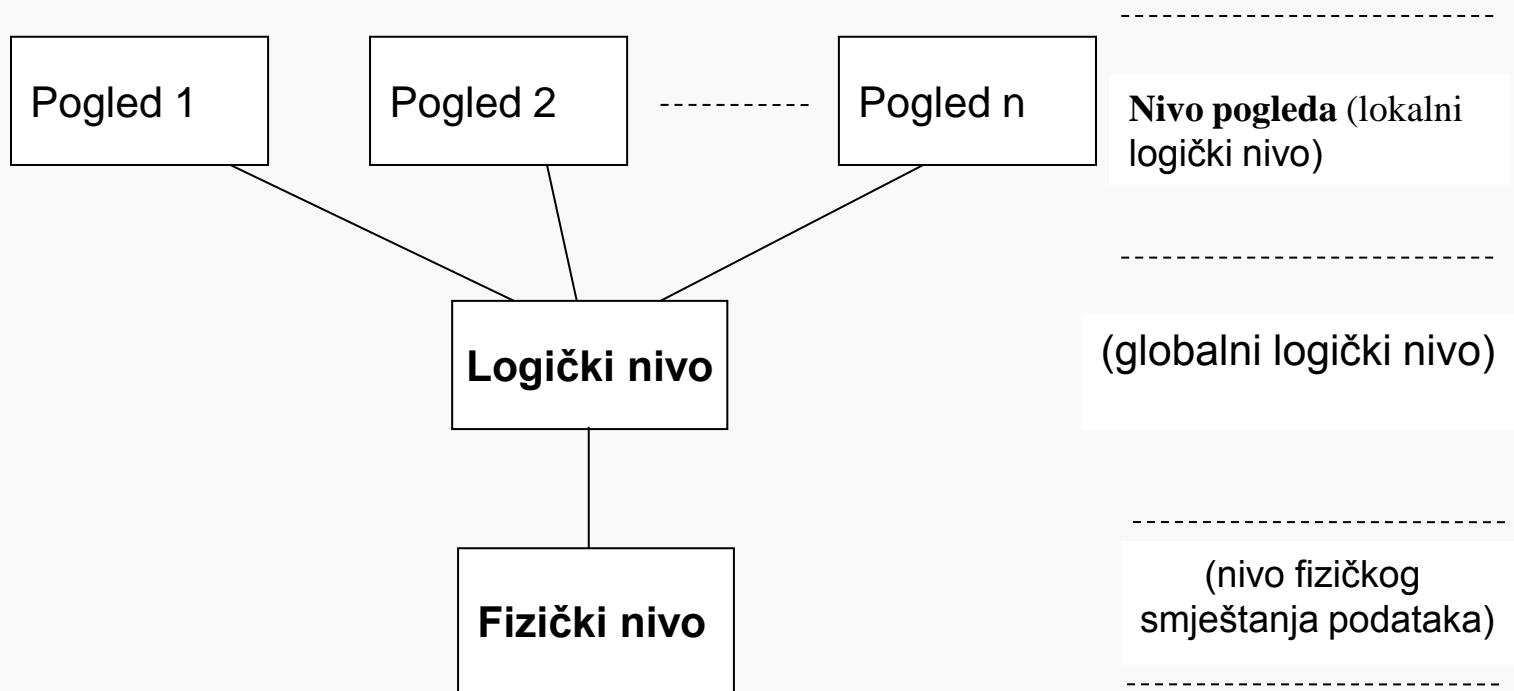
Specifikacija strukture podataka na logičkom nivou odgovara specifikaciji tipova promjenljivih u programskim jezicima

Nivoi Apstrakcije Podataka u Sistemima sa BP

PREDSTAVA PODATAKA

- **Nivo pogleda** : korisniku su dostupni samo podaci koji su mu potrebni (kroz aplikacioni program, koji dodatno može da oslobodi korisnika potrebe za poznavanjem tipa podataka, ili kroz specifikaciju pogleda preko kojih se specifikuju podaci dostupni korisniku). Pogledi omogućavaju skrivanje određenih informacija od korisnika iz razloga sigurnosti.

TRI NIVOVA APSTRAKCIJE PODATAKA (Ansi Sparc arhitektura)



Modeli podataka

- Na svakom nivou apstrakcije (posmatranja) podataka, potrebno je na neki način dati opis/predstavu podataka.

- Kolekcija koncepata koja se koristi za opis

- ✎ strukture podataka u bazi
- ✎ relacija između podataka
- ✎ semantike
- ✎ ograničenja

naziva se **model podataka**.

- **Konceptualni modeli** ili **modeli visokog nivoa** (eng. *high level models*) su modeli koji koriste koncepte kojima se opisuju podaci u bazi, njihove veze itd. na način blizak korisniku,

- ✎ Najčešće korišteni model visokog nivoa, je Model Objekti-Veze (Entity-Relationship model)

Ovaj logički model podataka predstavlja opis podataka i veza između njih na najvišem nivou apstrakcije

- **Fizički modeli podataka** ili **modeli niskog nivoa** (eng. *low level models*).

- ✎ Koriste se za opis-specifikaciju organizacije i načina fizičkog smještanja i podataka na sekundarnim memorijskim medijima,

Modeli podataka

■ Reprezentacioni (ili implementacioni) modeli podataka,

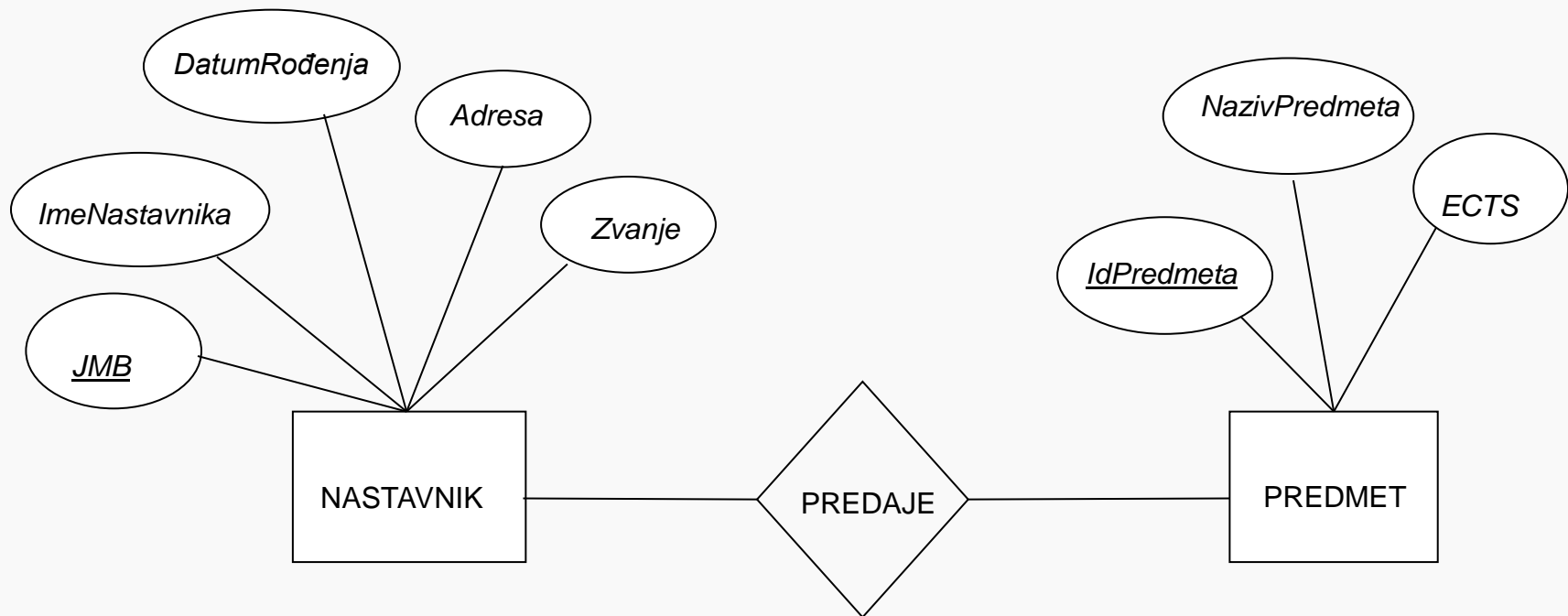
- 👉 Klasa modela , koji sadrže koncepte za opis podataka koji su razumljivi korisnicima, ali koji reflektuju način na koji su podaci organizovani u računarskoj bazi. Primjeri:
- 👉 **relacioni model**, najpoznatiji i najčešće korišten reprezentacioni model je koji odražava način organizacije podataka u relacionim bazama podataka.
- 👉 **hijerarhijski i mrežni model**, zastarjeli modeli za reprezentaciju logičke organizacije podataka u hijerarhijskim odnosno mrežnim bazama podataka.
- 👉 Prethodni modeli sveukupnu predstavu organizacije podataka opisuju koristeći logičke zapise, kao osnovne strukture iz kojih se izgrađuje model, pa se iz tog razloga ovi modeli nazivaju **modeli podataka zasnovani na zapisima**.

■ Drugi modeli:

- 👉 objektno-orijentisani model
- 👉 djelomično-strukturirani modeli podataka (**XML opisi podataka; podaci mogu imati različit broj atributa**)

Model Objekti-Veze

Primjer specifikacije strukture podataka, njihovih veza, značenja i ograničenja u modelu objekti-veze



Model Objekti – Veze (Nast.)

- MOV (E-R) modelom se modeluje – opisuje realni sistem.
Elementi modela:
 - 👉 Entiteti (reprezentuje objekte realnog svijeta)
 - 📄 Npr. Nastavnik 'Marić', student 'Perić', predmet 'Baze podataka'
 - 👉 Veze između entiteta – objekata realnog svijeta
 - 📄 Npr. Nastavnik 'Marić' predaje predmet 'Baze Podataka'
 - 👉 Vezni i entitetski skupovi i tipovi. Npr. Vezni tip/skup **PREDAJE** povezuje odnosne entitetske tipove/skupove **NASTAVNIK** i **PREDMET**, odnosno predstavlja skup (mogućih/konkretnih) veza između **nastavnika** i **predmeta** koje nastavnici predaju.
- MOV model je široko rasprostranjen i najčešće se koristi za konceptualno modelovanje baza podataka
 - 👉 Polazeći od MOV modela neke baze podataka, može se konverzijom po određenim pravilima doći do relacionog modela (šeme) koja opisuje sveukupnu logičku strukturu relacione baze podataka.

Relacioni Model

- Primjer tabelarne predstave podataka u relacionom modelu

STUDENT

ImeStudenta	JMB	GodinaStudija	StudijskiProgram
Jovanović	12109 87 100018	3	RI
Marković	05129 83 100067	4	RI
Petrović	22089 88 105034	2	ET
Živanović	06079 89 100021	2	ET
...

Jednostavna Relaciona Baza Podataka

STUDENT	ImeStudenta	JMB	GodinaStudija	StudijskiProgram
	Jovanović	12109 87 100018	3	RI
	Marković	05129 83 100067	4	RI
	Petrović	22089 88 105034	2	ET
	Živanović	06079 89 100021	2	ET

PREDMET	NazivPredmeta	IdPredmeta	ECTS
	Ekspertni sistemi	ES100	6
	Baze podataka	BP101	6
	Računarske mreže	RM102	6
	Internet Programiranje	IP100	7

POLAGANJE ISPITA	JMB	IdPredmeta	DatumIspita	Ocjena
	05129 83 100067	ES100	12/10/2009	7
	12109 87 100018	BP101	14/06/2011	9
	05129 83 100067	RM102	22/02/2010	9
	05129 83 100067	BP101	13/07/2010	8

Šeme i Instance

Analogija sa pojmovima deklaracije i sadržaja promjenljivih u programskim jezicima

- **Šema baze podataka** – predstavlja sveukupan opis (dizajn, specifikaciju) konkretne baze podataka

Šema korespondira deklaracijama promjenljivih u programskim jezicima.

Sistem sa bazom podataka može imati nekoliko šema baza podataka, prema različitim nivoima apstrakcije

- 👉 **Fizička šema:** opis (dizajn, specifikacija) BP na fizičkom nivou
- 👉 **Logička šema:** opis (dizajn, specifikacija) BP na implementacionom logičkom nivou
- 👉 **Konceptualna šema** opisuje sveukupnu logičku strukturu podataka na visokom nivou apstrakcije.
- 👉 **Podšema:** dio logičke šeme BP na nivou pogleda

- **Instanca** –aktuelni sadržaj baze podataka u određenom trenutku vremena

👉 Analogija sa vrijednostima promjenljivih u nekom trenutku vremena

- **Fizička nezavisnost podataka** – mogućnost promjene fizičke šeme (specifikacije fizičke organizacije) bez potrebe za promjenom logičke šeme




- 👉 Aplikacije koriste specifikacije logičke (implementacione) šeme
- 👉 Uopšteno, interfejsi između različitih nivoa i komponenata trebaju biti standardizovani i dobro koncipirani, tako da promjene u jednom dijelu ne utiču značajnije na ostale dijelove – komponente sistema.

Jezik za Definisanje Podataka (DDL)

- Primjer notacije za definisanje šeme baze podataka sistemu za upravljanje relacionom bazom podataka

 Primjer :

```
create table STUDENT (  
    ImeStudenta varchar(50),  
    JMB numeric (13,0),  
    GodinaStudija integer,  
    StudijskiProgram varchar(10) ) ;
```

- DDL kompajler generiše skup tabela čiji se opis smješta u riječnik podataka (*data dictionary*)
- *Data dictionary* sadrži metapodatke (tj. podatke o podacima)
 -  šema baze podataka (database schema) – primjer metapodataka
- *Data storage and definition language* (specijalni tip DDL)
 -  Jezik kojim se opisuje organizacija i način smještanja podataka na fizičkom mediju te pristupni metod
 -  Obično ekstenzija jezika za definiciju podataka (DDL)

Jezik za Manipulaciju Podacima (DML)

- Jezik za pristup i manipulaciju podacima organizovanim po određenom modelu podataka
 - 👉 Za DML (data manipulation language) se takođe koristi termin ***upitni jezik (query language)***
- Dvije klase DML
 - 👉 Proceduralni – korisnik specifikuje koje podatke želi i kako izvršiti ekstrakciju podataka iz baze
 - 👉 Neproceduralni – korisnik specifikuje koje podatke želi bez specifikacije kako izvršiti ekstrakciju podataka
- SQL je najčešće korišteni, neproceduralni upitni jezik

SQL

- SQL: široko rasprostranjen i korišten
- Npr. imena studenata treće godine studija možemo dobiti specifikacijom sljedećeg SQL upita:

```
select ImeStudenta  
from STUDENT  
where GodinaStudija = 3
```

- Aplikativni programi generalno pristupaju podacima u bazi na jedan od sljedećih načina
 - 👉 Kroz extenzije u programskim jezicima koje omogućavaju 'ugrađene' SQL iskaze.
 - 👉 API (application program interface) (e.g. ODBC/JDBC) koji omogućava da se SQL upiti proslijede DBMS.



Nedostaci file baziranih sistema sa BP i razlozi nastanka DBMS

- U početnom periodu razvoja, aplikacije koje su radile sa bazama podataka su koristile file sisteme za smještanje i pristup podacima.
- Nedostaci korištenja file sistema za smještanje podataka su:
 - ☞ Redundantnost i nekonzistentnost podataka
 - 📄 Isti podaci su smješteni višestruko (od strane različitih programa), u različitim fajlovima, sa različitom strukturom zapisa.
 - ☞ Ekstrakcija podataka iz baze nije jednostavna (*data isolation*) - podaci kojima je aplikativni program pristupao su se nalazili u različitim fajlovima sa različitim formatima – strukturom zapisa.
 - ☞ Teškoće u dobijanju željenih podataka i izvještaja
 - 📄 Bila je potrebna modifikacija programa ili novi program za dobijanje modifikovanih/novih izvještaja
 - ☞ Problem integriteta
 - 📄 Ograničenja za integritet podataka (npr. $ECTS > 0$) je ugrađivan u programski kod.
 - 📄 Nije bilo jednostavno dodavati nova ograničenja ili mijenjati postojeća



Nedostaci file baziranih sistema sa BP i razlozi nastanka DBMS

■ Nedostaci u korištenju file sistema (nast.)

Atomičnost ažuriranja

-  Otkazi u radu sistema su mogli ostaviti bazu u nekonzistentnom stanju sa parcijalno izvršenim ažuriranjem
-  Npr. transfer novca sa jednog računa na drugi, mora se ili kompletirati ili u potpunosti stornirati.

Istovremeni pristup od strane više korisnika

-  Konkurentnost je potrebna zbog postizanja sveukupno većih sistemskih performansi i brzog odziva
-  Nekontrolisan konkurentni pristup može dovesti do nekonzistentnosti u bazi
 - Npr. dva korisnika koji istovremeno očitavaju i ažuriraju stanje računa

Problem sigurnosti

■ DBMS daju rješenja na sve gore navedene probleme

Atomičnost ažuriranja

T_1

read [X_1]

$X_1 = X_1 - 400$

write [X_1]

read [X_2]

$X_2 = X_2 + 400$

write [X_2]

Transakcijom se prebacuje iznos (400 KM) sa računa X1 na račun X2

Problem?

Konkurentno izvršavanje transakcija

Dvije transakcije skidaju iznos sa X1
X1 - stanje računa (1000KM)

T_1	T_2
read [X_1] $X_1 = X_1 - 300$	
write [X_1]	read [X_1] $X_1 = X_1 - 200$
	write [X_1]

**Stanje računa X1 nakon izvršenja
transakcija T1 i T2 ?**

- A) 200**
- B) 300**
- C) 500**
- D) 800**

Sistemi za Upravljanje Bazama Podataka

Ciljevi DBMS

- Aplikativna nezavisnost i primjenljivost za široki spektar aplikacija.
- Efikasno smještanje u bazu i pretraživanje i dobijanje podataka iz baze na način pogodan za korisnika.
 - 👉 DBMS sadrži **procesor upita** (eng. *query processor*) koji procesira i DDL i DML iskaze i proslijeđuje ih **podsystemu DBMS-a za upravljanje smještanjem podataka** (eng. *storage manager*).
Podsystem za upravljanje smještanjem podataka implementira efikasan fizički pristup podacima na disku i manipulaciju podacima.
- Podrška za tri nivoa apstrakcije podataka.
- Fizička i logička nezavisnost podataka i nezavisnost programa od podataka u sistemima koji koriste DBMS.
- Konkurentni rad više korisnika.
- Upravljanje autorizacijom korisnika i integritetom podataka

Sistemi za Upravljanje Bazama Podataka

Podrška za izvršavanje transakcija

- *Transakcija* je skup operacija koje obezbjeđuju izvršavanje neke logičke cjeline - funkcije u nekoj aplikaciji koja radi sa bazom podataka. ACID svojstva (Atomicity, Consistency, Isolation, Durability)
- ACID svojstva osigurava komponenta za upravljanje transakcijama (to je komponenta podsistema za smještanje podataka)
- Modul za upravljanje oporavkom osigurava da baza podataka ostaje u konzistentnom (korektnom) stanju i u slučaju sistemskih grešaka (npr. nestanak napajanja, padovi operativnog sistema, hardverski otkazi) ili kad se transakcija neuspješno završi (to je modul komponente za upravljanje transakcijama).
- Modul (komponente za upravljanje transakcijama) za upravljanje konkurentnim izvršavanjem kontroliše interakciju između konkurentnih transakcija, sa ciljem osiguranja konzistentnosti baze podataka.

Serijsko izvršavanje transakcija

Dvije transakcije skidaju (T1 i T2) iznose sa stanje računa (X1)

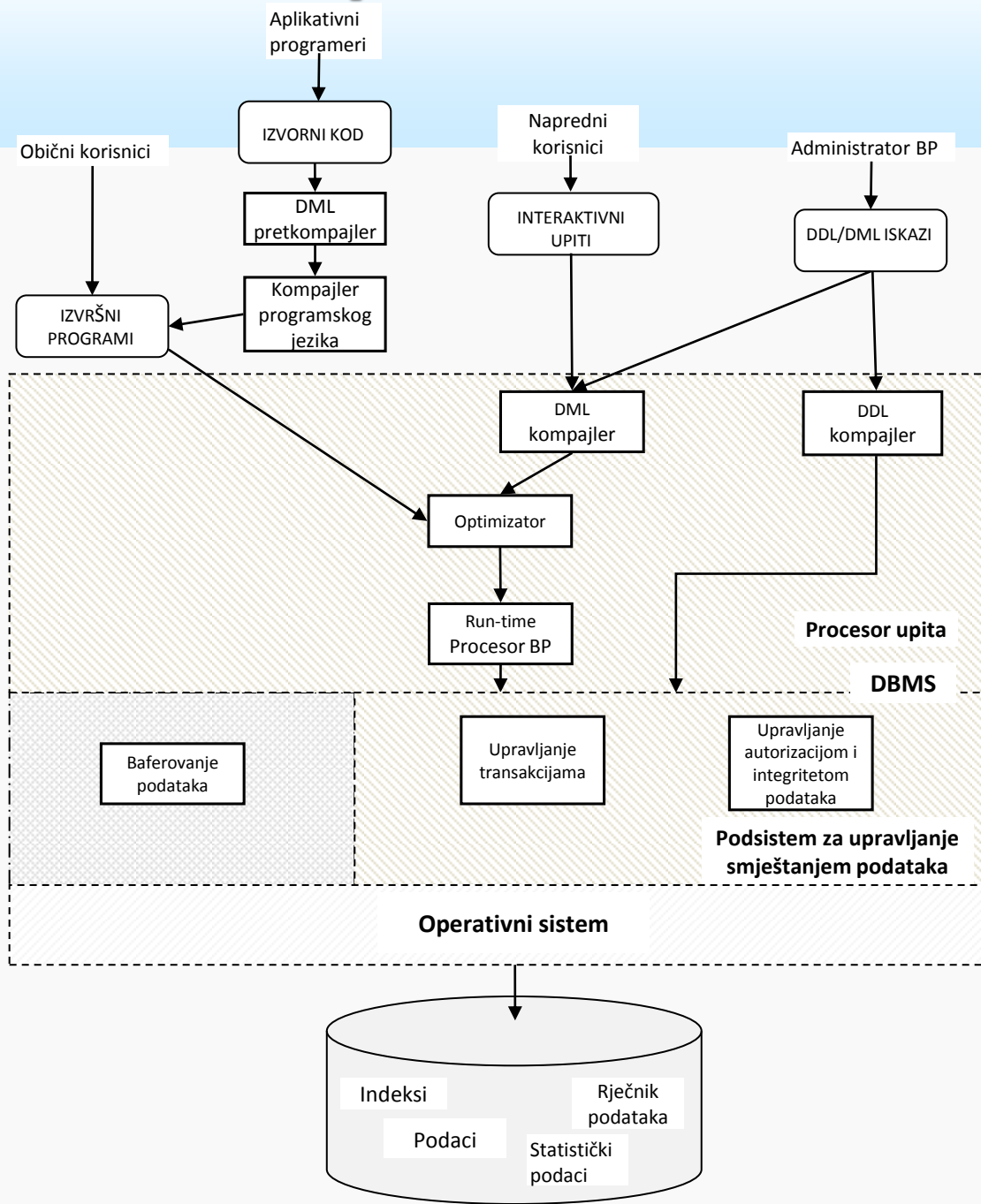
T_1	T_2
read [X_1] $X_1 = X_1 - 300$ write [X_1]	read [X_1] $X_1 = X_1 - 200$ write [X_1]

Sistemi za Upravljanje Bazama Podataka

Upravljanje smještanjem podataka – Storage Management

- Obezbeđuje interfejs između podataka smještenih na fizičkom mediju i procesora upita koji konvertuje upite iz aplikativnih programa/DML iskaza u instrukcije niskog nivoa.
- Storage manager konvertuje instrukcije niskog nivoa dobijene od procesora upita u operacije niskog nivoa za efikasan pristup i manipulaciju podacima u bazi.
- Storage manager je odgovoran za sljedeće zadatke:
 - ☞ Interakcija sa file managerom (npr. alokacija prostora na disku)
 - ☞ Efikasno smještanje, pretraživanje i ažuriranje podataka, upravljanje transakcijama, konkurentno izvršenje, autorizaciju i integritet
- Komponente Storage manager'a su:
 - ☞ Upravljanje baferom (Buffer manager), Komponenta za upravljanje transakcijama (Transaction manager), Komponenta za autorizaciju i osiguranje integriteta (Authorisation and Integrity Manager)
- Kreira i radi sa sljedećim strukturama:
 - ☞ Podaci / Datoteke
 - ☞ Riječnik podataka (Data dictionary)
 - ☞ Indeksne strukture i

Sveukupna Struktura Sistema



Korisnici Baza Podataka (Database Users)

- Korisnici se razlikuju po očekivanom načinu interakcije sa sistemom
- Obični korisnici – koriste neku od postojećih – prethodno realizovanih aplikacija
 - 👉 Npr. bankarski službenici, računovođe, radnici u matičnim uredima itd
- Aplikativni programeri – koriste različite programske jezike, RAD alate i generatore za kreiranje aplikacija
- Napredni (sofistikovani) korisnici (npr. analitičari podataka) – koriste upitni jezik za specifikaciju upita u bazu. Takođe koriste druge alate (OLAP, data mining)
- Administratori BP

Administrator Baze Podataka

- Koordinira sve aktivnosti vezane za bazu podataka; administrator baze podataka dobro poznaje informacione potrebe poslovnog sistema i informacije koje se vode u sistemu.
- Zadaci administratora baze podataka uključuju:
 - ✎ Definisanje logičke šeme baze podataka
 - ✎ Specifikacija integritetskih ograničenja
 - ✎ Definisanje fizičke organizacije i pristupnih metoda
 - ✎ Modifikacije šeme baze podataka i fizičke organizacije
 - ✎ Autorizacija korisnika za rad sa bazom podataka
 - ✎ Veza sa korisnicima, registrovanje njihovih zahtjeva, opažanja i poduzimanje adekvatnih akcija
 - ✎ Nadzor performansi i poduzimanje odgovarajućih akcija na osnovu zahtjeva korisnika

Klasifikacija DBMS

■ Prema tipu organizacije baze podataka koje DBMS podržava:

- 👉 **relacione,**
- 👉 **objektno-relacione,**
- 👉 **objektne,**
- 👉 **XML,**
- 👉 **hijerarhijske, mrežne**
- 👉 **NoSQL** itd.

■ Prema lokaciji baze podataka i DBMS:

- 👉 **centralizovane i**
- 👉 **distibuirane.**

■ Prema namjeni:

- 👉 **opšte i**
- 👉 **specijalne namjene.**

📖 *Stream DBMS, real-time DBMS, OLTP (eng. online transaction processing DBMS, npr. za aplikacije rezervacije avionskih karata) itd.*

Aplikativna Arhitektura

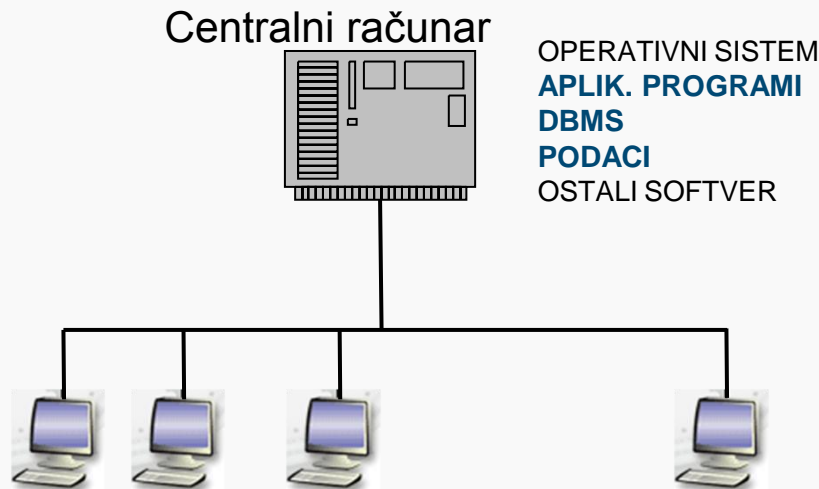
■ Aplikacija baze podataka se sastoji od

- ✎ Baze podataka
- ✎ programa za unos i ažuriranje, obradu, pretraživanje i prezentaciju podataka.

■ Arhitektura aplikacije:

- ✎ Centralizovana
- ✎ Klijent – server

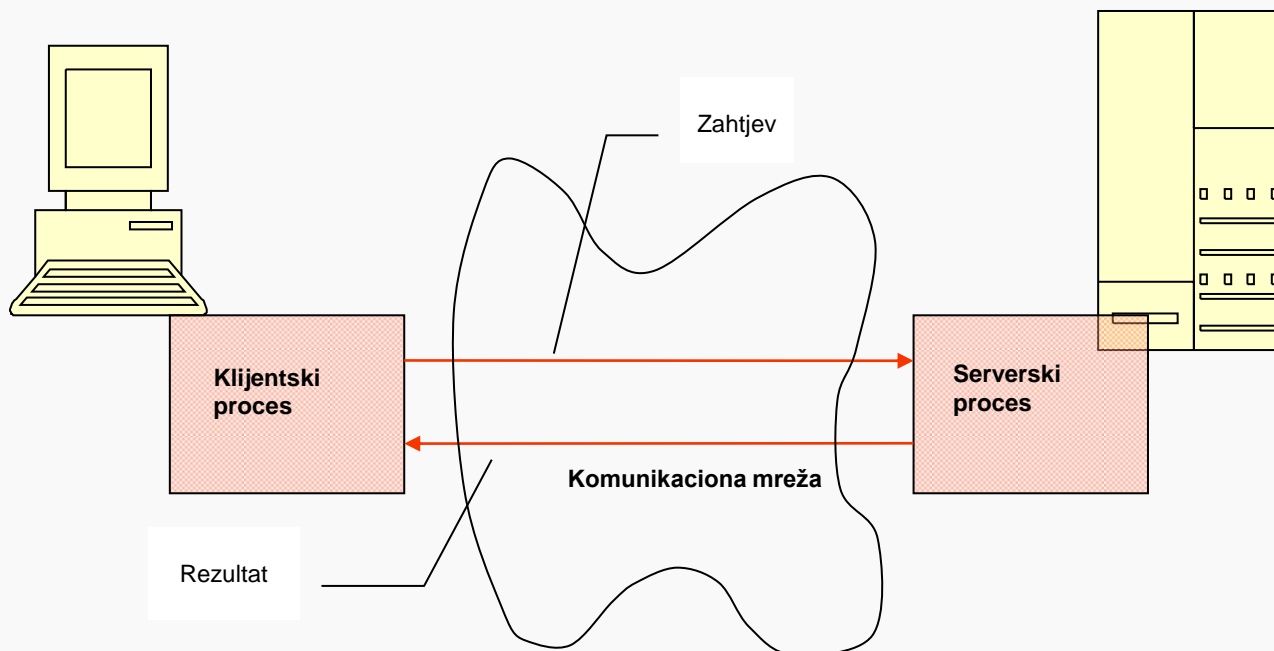
■ Centralizovana arhitektura



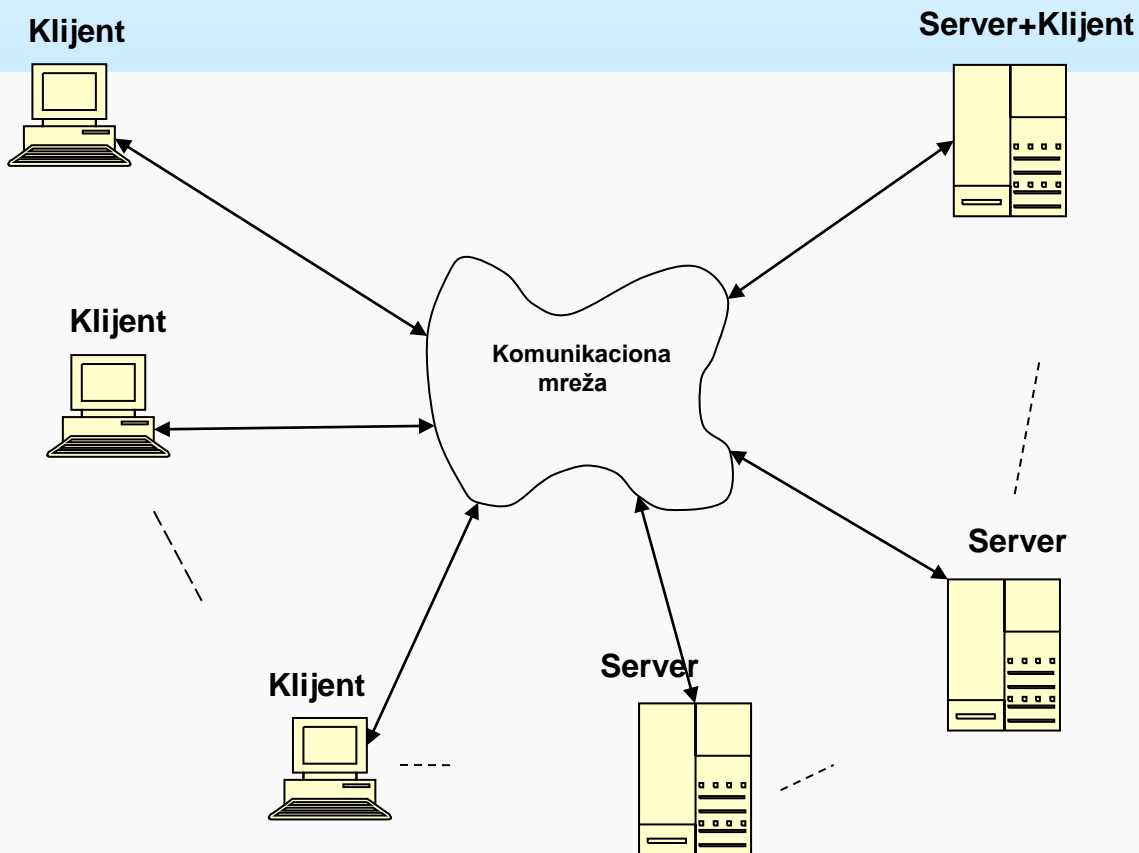
Aplikativna Arhitektura

■ Klijent- server arhitektura

☞ Princip klijent-server obrade

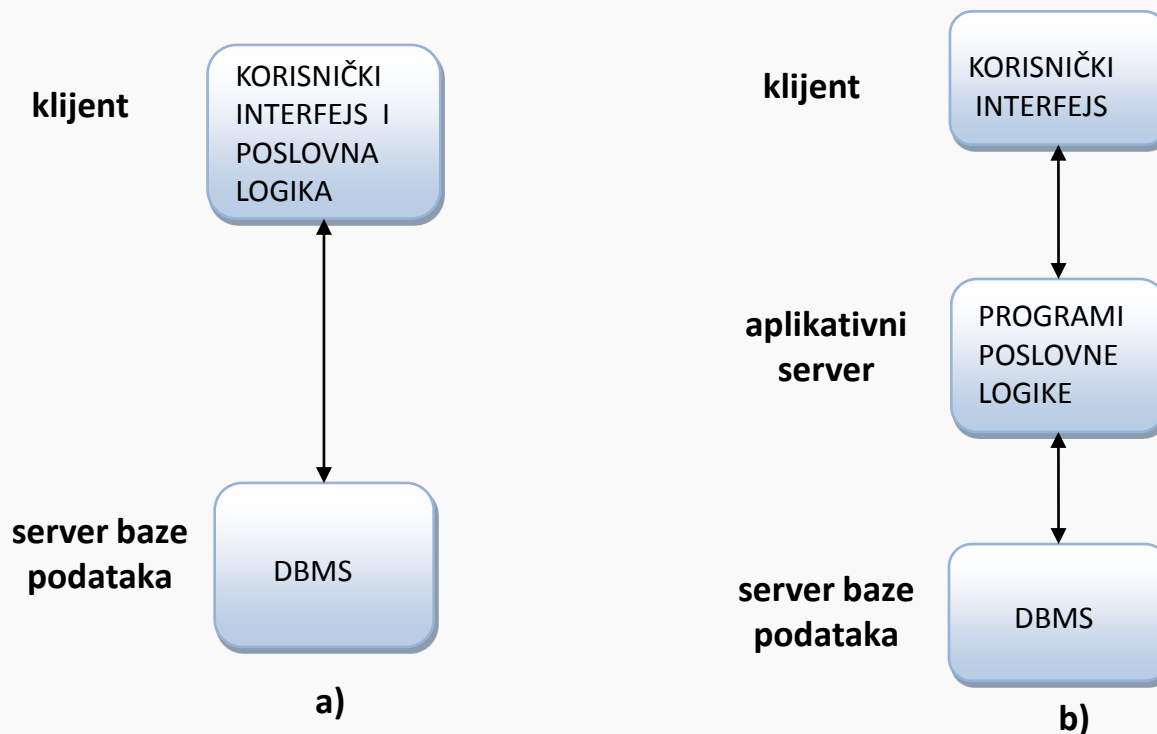


Aplikativna Arhitektura



Kompleksna klijent-server konfiguracija.

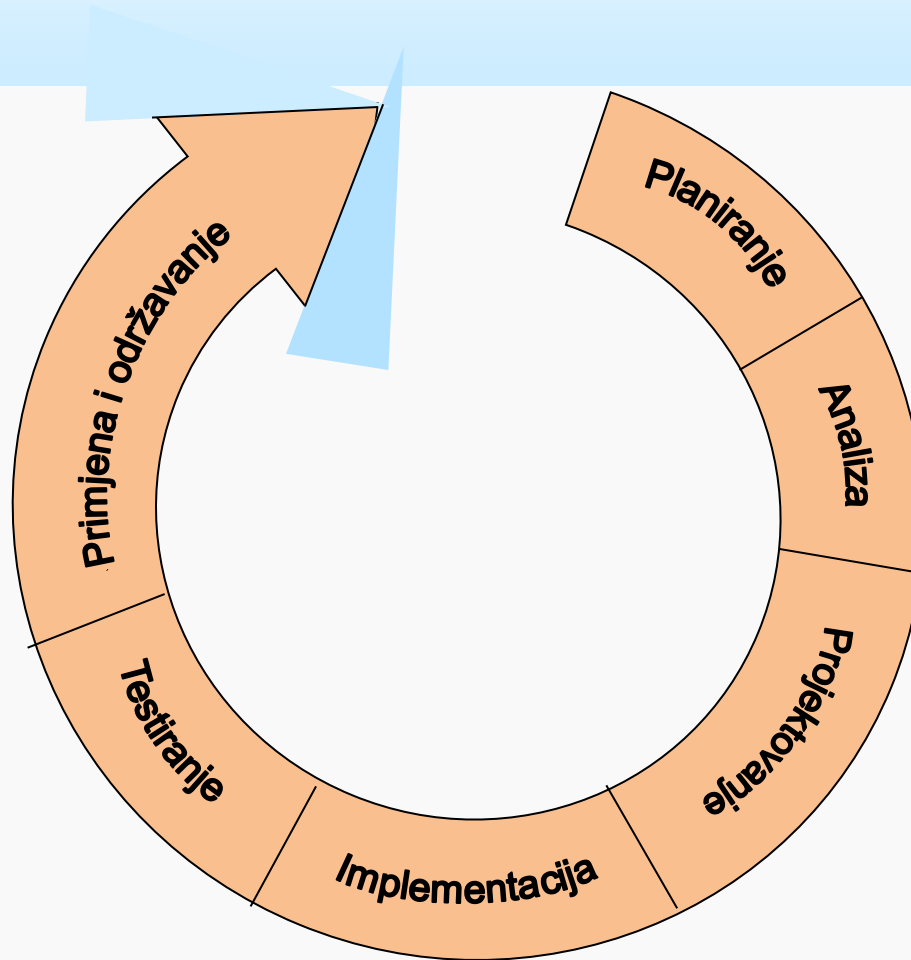
Aplikativna Arhitektura



a) Dvoslojna: Npr. klijentski program koristeći ODBC/JDBC komunicira sa bazom podataka

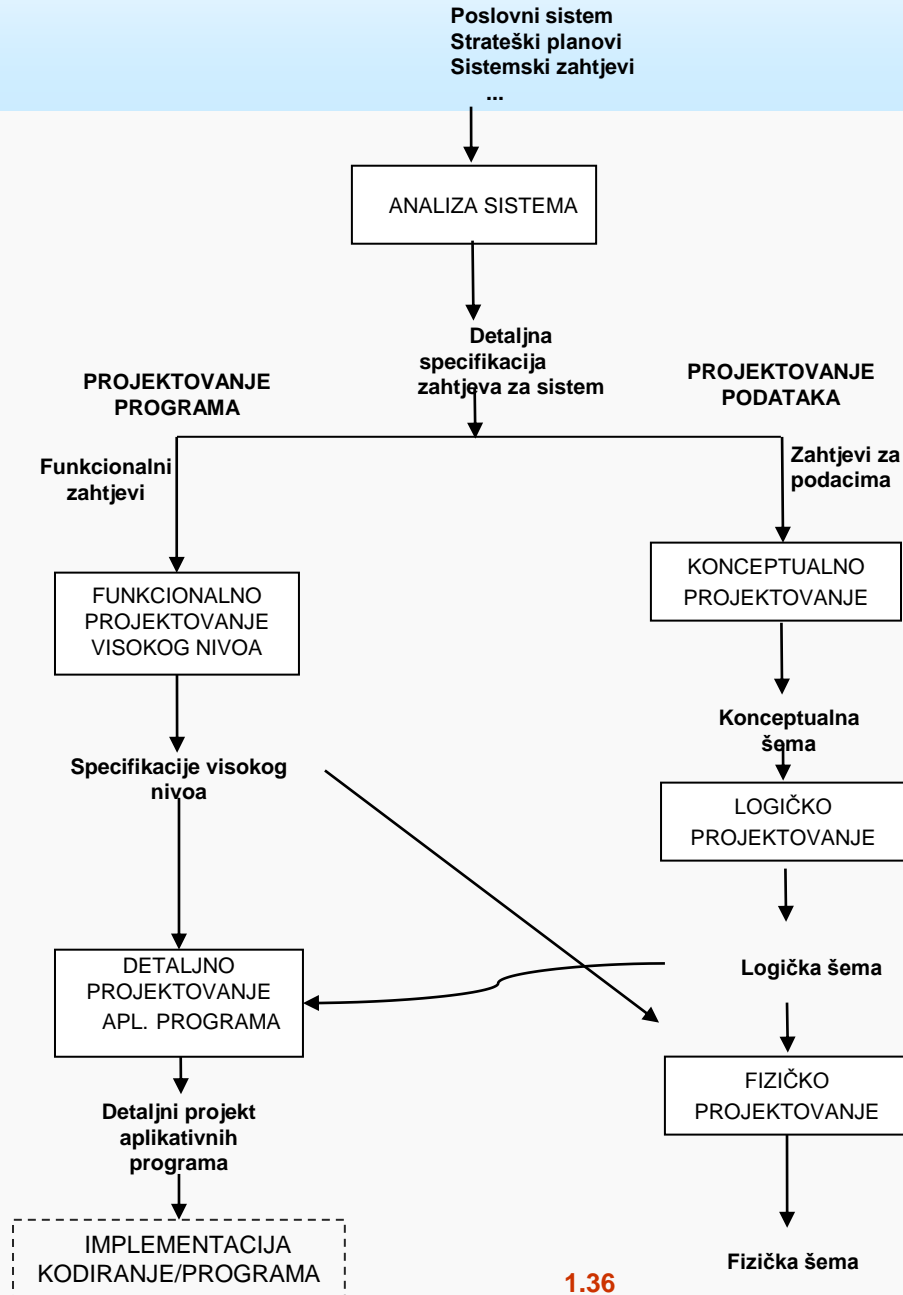
b) Troslojna arhitektura: Npr. web-bazirane aplikacije i aplikacije koje koriste međusloj za spregu (“middleware”)

Proces razvoja sistema sa bazom podataka



Životni ciklus sistema sa bazom podataka

Osnovne faze projektovanja baze podataka



Sažetak

- Produkcija informacija i potrebe za bazama podataka
- BP i Sistemi za Upravljanje Bazama Podataka
- Tipovi Baza podataka
- Pogled na podatke
- Modeli Podataka
- Šeme i instance
- Jezik za Definisanje Podataka (Data Definition Language)
- Jezik za Manipulaciju Podacina (Data Manipulation Language)
- Nedostaci file baziranih sistema sa BP i razlozi za uvođenje Sistema za Upravljanje Bazama Podataka
- DBMS: Ciljevi i osnovne komponente
- Klasifikacija DBMS
- Sveukupna Struktura DBMS Sistema
- Korisnici Baza Podataka
- Administrator Baze Podataka
- Aplikativna arhitektura