

BAZE PODATAKA

Uvod

BAZE PODATAKA – UVOD (1/3)

- Baza podataka se struktuirana kolekcija podataka koja postoji relativno dugo i koju najopštije može definisati kao dobro koristi i odžava, po pravilu, više korisnika, odnosno programa (aplikacija).
- Izučavanju baza podataka može se pristupiti sa dva različita, međusobno povezana aspekta u kojima se one tretiraju bilo kao:
 - Sistemi za upravljanje bazom podataka SUBP(Data Base Management System-DBMS)
 - Modeli podataka – intelektualni alata za projektovanje i razvoj BP


BAZE PODATAKA – UVOD (2/3)

- **Sistemi za upravljanje bazom podataka** (Database Management Systems), specifična tehnologija obrade podataka, odnosno softverski sistem koji obezbeđuje osnovne funkcije obrade velike količine podataka: jednostavno pretraživanje i održavanje podataka, višestruko paralelno (konkurentno) korišćenje istog skupa podataka, pouzdanost i sigurnost),
- **Modeli podataka**, odnosno specifične teorije pomoću kojih se specifikuje, projektuje i implementira neka konkretna baza podataka ili informacioni sistem, uopšte.

Sistemi za upravljanje bazom podataka obično se zasnivaju na nekom modelu podataka.

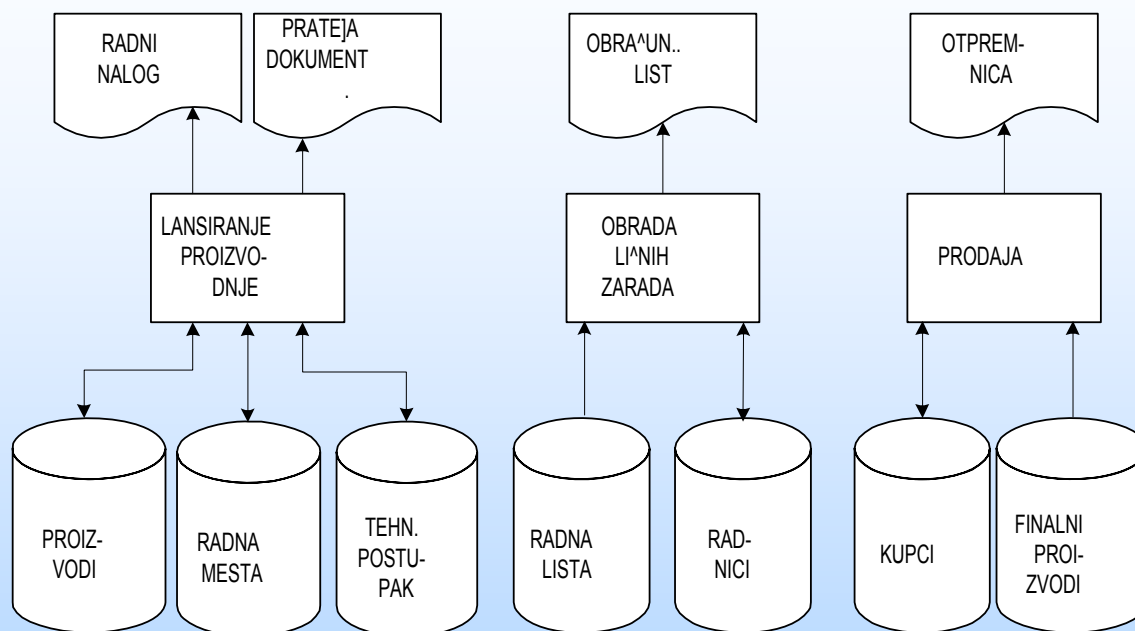
BAZE PODATAKA – UVOD (3/3)

- MODELI PODATAKA - PROJEKTOVANJE BAZA PODATAKA
- SISTEMI ZA UPRAVLJANJE BAZOM PODATAKA- RAZOJ
SOFTVERA U OKRUŽENJU BAZA PODATAKA



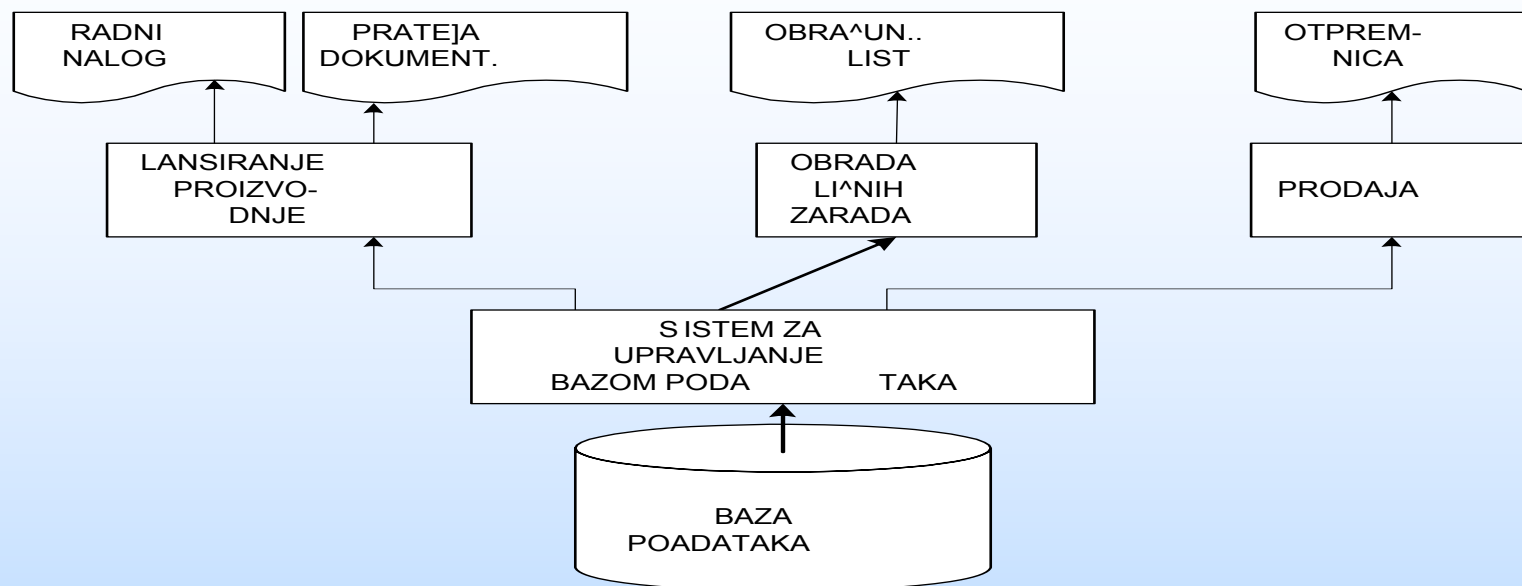
BAZE PODATAKA -SISTEMI ZA UPRAVLJANJE BAZOM PODATAKA (SUBP)

KONVENCIONALNA OBRADA - SISTEM DATOTEKA



- Redudansa podataka
- Logička i fizička zavisnost podataka i programa
- Niska produktivnost u razvoju sistema
- Nezadovoljavajuće pouzdan (otkazi sistema)
- Ne obezbedjuje tačnost i konzistentnost pri paralelnom radu

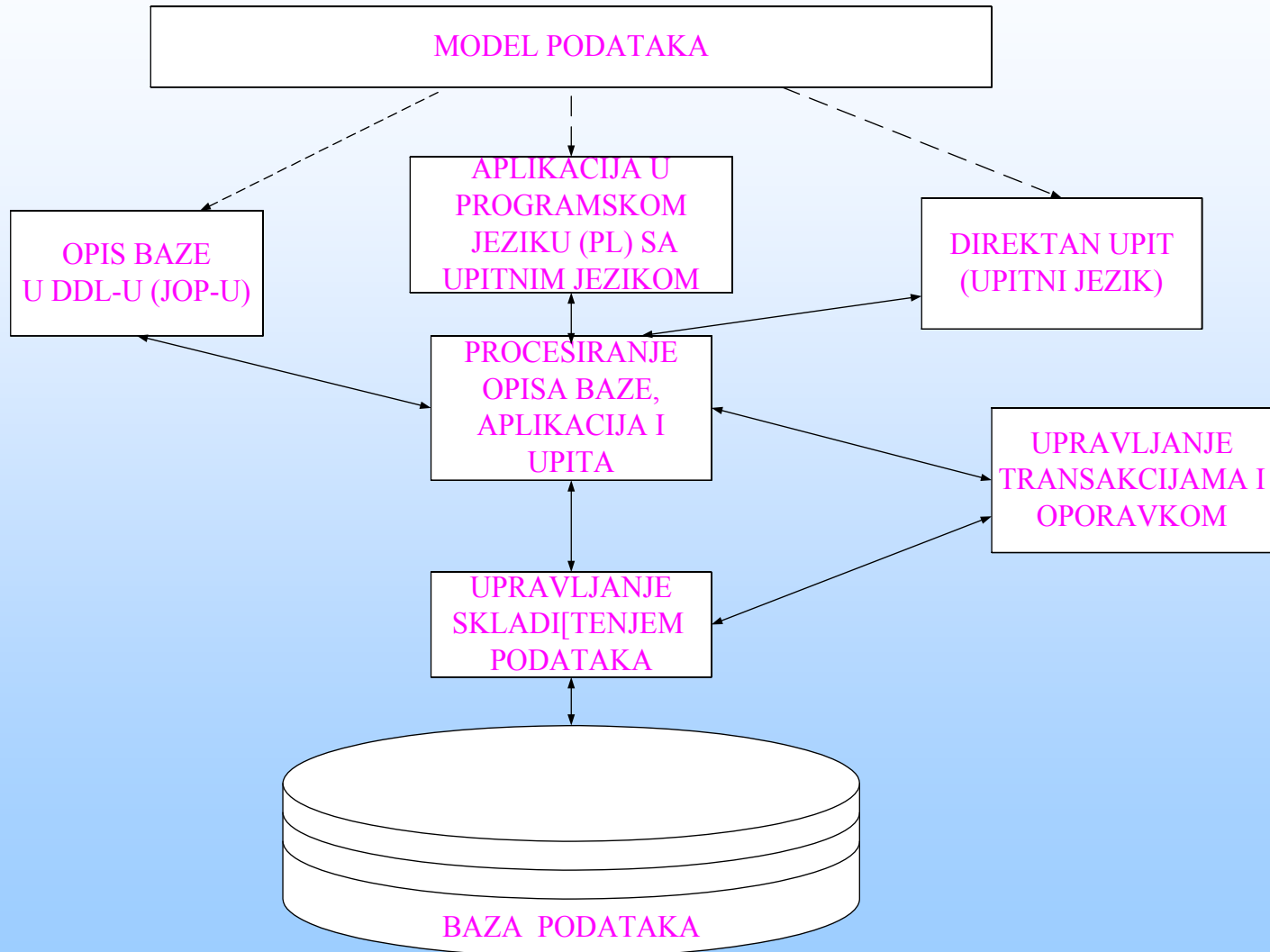
SISTEM ZA UPRAVLJANJE BAZOM PODATAKA –SUBP



Skladištenje podataka sa minimumom redundanse;

- Pouzdanost podataka i pri mogućim hardverskim i softverskim otkazima;
- Pouzdano paralelno korišćenje zajedničkih podataka od strane više ovlašćenih korisnika;
- Logičku i fizičku nezavisnost programa od podataka.
- Jednostavnu komuniciranju sa bazom podataka preko jezika bliskih korisniku, tzv "upitnih jezika"

KOMPONENTE SUBP-a



BAZA PODATAKA (1/2)

- Velike baze podataka pored diskova (sekundarne memorije) zahtevaju i tzv "tercijalnu" memoriju. Jedinice tercijalne memorije imaju kapacitet reda terabajta (1000 gigabajta, odnosno 10^{12} bajta). Na primer, sistem kompakt diskova sa "robotom" za izbor konkretnog diska. Očigledno je da je pristup podacima na tercijalnoj memoriji znatno sporiji (nekoliko sekundi) od pristupa podacima na diskovima (10 -20) milisekundi.
- Sistem za upravljanje skladištenjem podataka mora da obezbedi jednoobrazan pristup podacima na svim vrstama memorije.

BAZA PODATAKA (2/2)

- Baza podataka, pored podataka, sadrži i ***metapodatke***, odnosno tzv "***Rečnik podataka***" (Data Dictionary, Data Directory, Catalog). ***Rečnik baze podataka*** opisuje posmatranu bazu podataka (strukturu baze, pravila očuvanja integriteta podataka, prava korišćenja i slično). Rečnik podataka je "baza podataka o bazi podataka", pa se taj deo baze podataka naziva i ***metabaza*** podataka.
- SUBP održava i ***bazu indeksa***. ***Index***, najopštije, predstavlja strukturu podataka koja omogućava brz pristup "indeksiranim" podacima baze. Najčešća struktura indeksa je B-stablo. Rečnik podataka sadrži i opis indeksa posmatrane baze.

SISTEM ZA UPRAVLJANJE SKLADIŠTENJEM PODATAKA

- Sadrži dve osnovne komponente, *Upravljanje baferima* (Buffer Manager) i *Upravljanje datotekama* (File Manager).
- **Upravljanje datotekama** vodi računa o lokaciji datoteka i o pristupima blokovima (4.000 do 16.000 bajta) podataka na zahtev Upravljanja baferima.
- **Upravljanje baferima** prihvata blok podataka sa diska, dodeljuje mu izabranu stranicu centralne memorije, zadržava ga izvesno vreme, u skladu sa ugrađenim algoritmom upravljanja baferima, a zatim vraća na disk oslobađajući dodeljenu mu stranicu.

ODRŽAVANJE ŠEME BAZE PODATAKA

- Održavanje šeme (opisa) baze podataka. Šema baze podataka opisuje strukturu baze podataka, pravila integriteta i prava korišćenja . Održavanje šeme baze podataka podrazumeva prvo kreiranje, a zatim i modifikovanje ovog opisa koji se čuva u Rečniku podataka
- **Data Definition Language - DDL** (Jezik za opis podataka - JOP) koji se koristi za održavanje šeme baze podataka - naziv za konvencionalne baze
- Object definition language - ODL za objektne baze

UPITI I UPITNI JEZICI (1/2)

- **Upitni jezici -Neproceduralni jezici** sadrže konstrukcije preko kojih se samo specificuju uslovi koje treba da zadovolji željeni rezultat, a ne i procedura pomoću koje se dobija taj rezultat.
- Osnovni zadatak **Procesora upita (Query Processor)** je da transformiše neproceduralni iskaz u sekvencu zahteva koje treba da realizuje **Sistem za upravljanje skladištenjem podataka**
- **Optimizacija upita** je nalaženje najpogodnije procedure za realizaciju neproceduralnog iskaza. Optimizacija upita koristi podatke iz Rečnika podataka: opis strukture, pravila integriteta, prava pristupa i definiciju indeksa.

UPITI I UPITNI JEZICI (2/2)

- **SQL:1999 (Structured Query Language)** - standardni upitni jezik za relacione baze podataka (i objektno-relacione)
- **OQL (Object Query Language)** - standardni upitni jezik za objektno baze podataka
- **SQL:1999 = SQL3**
- **SQL92 = SQL2** prethodni standard

DATA MANIPULATION LANGUAGE - DML

- *Jezik za manipulaciju podataka* - JMP (Data Manipulation Language - DML) je opšte ime za jezik preko koga se pristupa podacima u bazi i oni čitaju i menjaju.
- U relacionim bazama podataka DML je (kao i DDL) SQL.
- U starijim bazama (mrežnom i hijerarhijskom) postojali su specifični DML koji su praktično bili ugrađeni u programski jezik ("jezik domaćin").
- Ista strategija se koristi i u objektnim bazama, s tim što je ovde i ODL DML

OKRUŽENJE ZA RAZVOJ APLIKACIJA (1/2)

- I. **Jezici IV generacije - generatori aplikacija** (Relacioni model). Generišu se aplikacije na osnovu sličnosti struktura dela baze(tabela) i korisničkog interfejsa. Dvoslojna softverska arhitektura - čvrsta veza baze i korisničkog interfejsa. Problemi koji odatle proizilaze su značajni.
- II. **Aplikacija** se razvija u nekom programskom jeziku ("jeziku domaćinu") u koga se na neki način ugrađuje DML.

OKRUŽENJE ZA RAZVOJ APLIKACIJA (2/2)

- Bitno se razlikuju načini ugrađivanja DML-a u relacionim i objektnim bazama.
- U relacionim bazama postoji tzv" impedance mismatch" jezika domaćina i SQL-a. Cilj je da se ostvari nezavisnost jezika domaćina i upitnog jezika.
- U objektnim bazama DML se potpuno prilagođava programskom jeziku. Zato postoje C++ Binding, Java Binding, Smalltalk Binding.

DISTRIBUIRANE BAZE I RAZNE VRSTE KLIJENT-SERVER OKRUŽENJA

- Distribuirane baze – ostvarivanje transparentnosti distribucije
- Klijent-server okruženja:
 - Ostvarivanje konekcije sa BP
 - Call-Level Interface (CLI)
 - ODBC
 - JDBC
 - Različite vrste "middleware" – složenih komponenti za komunikaciju sa distribuiranim objektima (CORBA, .NET, EJB)

TRANSAKCIJA (1/4)

- **Transakcija** je niz operacija nad bazom podataka koji odgovara jednoj logičkoj jedinici posla u realnom sistemu.

*Učitaj **iznosp** za prenos;*

*Nađi račun R1 sa koga se **iznosp** skida;*

*Upiši **iznosR1 - iznosp** na račun R1;*

*Nađi račun R2 na koga se **iznosp** stavlja;*

*Upiši **iznosR2 + iznosp** na račun R2.*

- **Transakcija** u izvršenju mora da ima tzv. **ACID osobine** (po početnim slovima sledećih engleskih reči):

TRANSAKCIJA (2/4)

- ***Atomnost*** (Atomicity). Zahteva se da se bilo sve operacije nad bazom podataka uspešno obave ili ne obavi nijedna. Da bi se ostvarila atomnost transakcije definišu se dve specifične operacije nad bazom podataka:
 - COMMIT koja označava uspešan kraj transakcije i koja "potvrđuje" sve promene u bazi koje je posmatrana transakcija proizvela;
 - ROLLBACK kojom se poništavaju efekti svih prethodnih operacija nad bazom podataka u jednoj transakciji, ako ona, zbog predviđene ili nepredviđene greške (otkaza sistema) može da dovede bazu podataka u nekonzistentno stanje.

TRANSAKCIJA (3/4)

- **Konzistentnost** (Consistency). Očigledno je da se transakcija može definisati i kao "jedinica konzistentnosti" baze podataka: pre početka i posle okončanja transakcije stanje baze podataka mora da zadovolji uslove konzistentnosti. Za vreme obavljanja transakcije konzistentnost baze podataka može da bude narušena.

TRANSAKCIJA (4/4)

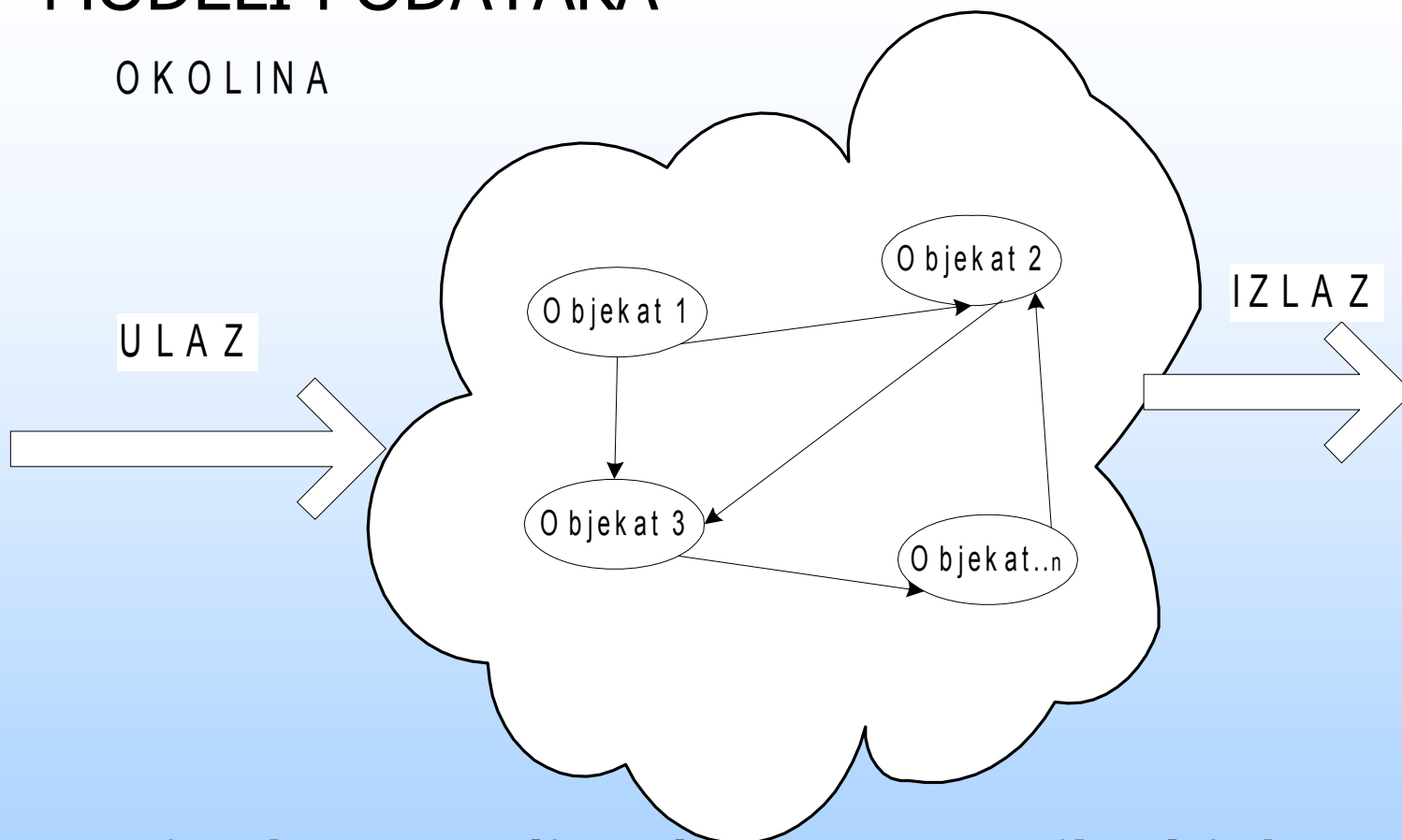
- ***Izolacija*** (Isolation). Kada se dve ili više transakcija izvršavaju istovremeno, njihovi efekti moraju biti međusobno izolovani. Drugim rečima efekti koje izazovu transakcije koje se obavljaju istovremeno moraju biti jednaki efektima nekog njihovog serijskog (jedna posle druge) izvršenja.
- ***Trajnost*** (Durability). Kada se transakcija završi njeni efekti ne mogu biti izgubljeni, čak i ako se neposredno po njenom okončanju desi neki ozbiljan otkaz sistema.



MODELI PODATAKA

MODELI PODATAKA

OKOLINA



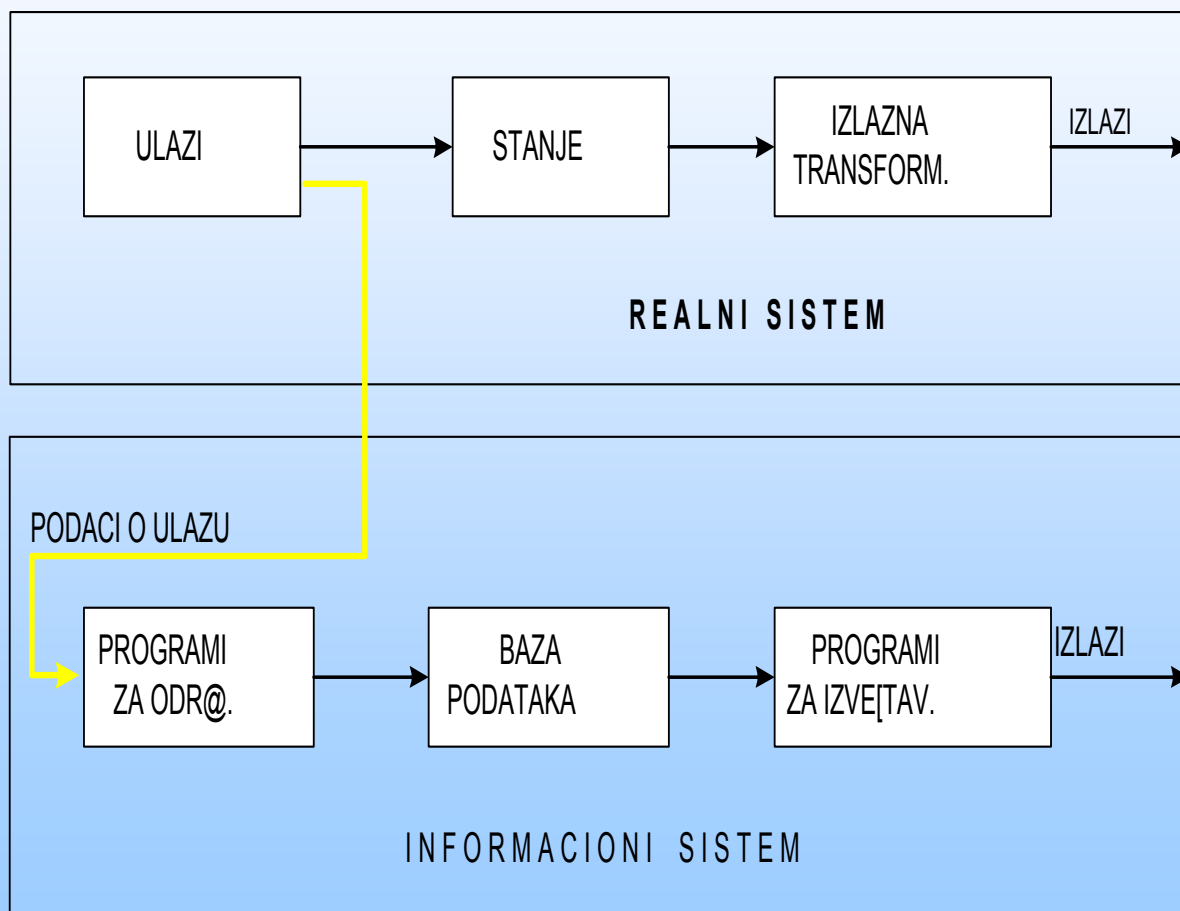
Sistem je skup medjusobno povezanih objekata. Ulazi u sistem opisuju dejstvo okoline na sistem, a izlazi odgovarajući odziv sistema, odnosno dejstvo sistema na okolinu.

MODELI PODATAKA

- Sistem, odnosno objekti imaju svoja stanja koja se menjaju pod dejstvom ulaza, a čija se promena reflektuje na izlaz.
- **Stanje sistema** predstavlja skup vrednosti atributa njegovih objekata i skup veza između objekata u datom trenutku vremena.

MODEL PODATAKA

INFORMACIONI SISTEM KAO MODEL REALNOG SISTEMA



MODEL PODATAKA - STANJE SISTEMA

- Ulazi u sistem (dejstvo okoline na sistem) menjaju stanja sistema
- Stanje sistema se definiše kao skup informacija o prošlosti i sadašnjosti sistema koji je potreban da bi se, pod dejstvom budućih poznatih ulaza, mogli odrediti budući izlazi. U stanju sistema skoncentrisana je celokupna istorija realnog sistema.
- Očigledno je da stanje sistema opisuje fundamentalne karakteristike sistema. U jednom trenutku vremena ono predstavlja skup objekata sistema, skup njihovih međusobnih veza i skup vrednosti atributa objekata u tom trenutku vremena.

MODEL PODATAKA

- Stanje sistema ne mora predstavljati neku "fizičku veličinu", odnosno neki poznati koncept realnog sistema - ono je prosto "skup informacija".
- Ponašanje sistema se u realnom sistemu sagledava ("meri"), odnosno sistemom se upravlja na bazi njegovih stvarnih izlaza. Izlazna transformacija definiše odnos između stanja sistema i izlaza, odnosno predstavlja način merenja ili posmatranja dinamičkog ponašanja realnog sistema.

MODEL PODATAKA - STANJE SISTEMA

- BAZA PODATAKA PREDSTAVLJA MODEL STANJA SISTEMA.
- **Model podataka** je intelektualni alat za definisanje modela sistema, za prikazivanje objekata sistema, njihovih atributa i njihovih dozvoljenih vrednosti, medjusobnih veza objekata i dinamike sistema.
- **Model podataka** je specifičan teorijski okvir pomoću koga se specifikuje, projektuje i implementira neka konkretna baza podataka ili informacioni sistem, uopšte.
- **Model podataka** je osnova za razvoj Sistema za upravljanje bazom podataka (SUBP)

MODEL PODATAKA

- Podatak je neka kodirana činjenica iz realnog sistema, on je nosilac informacije. Informacija je protumačeni (interpretirani) podatak. Interpretacija podataka se vrši na osnovu strukture podataka, semantičkih ograničenja na njihove vrednosti i preko operacija koje se nad njima mogu izvršiti. Imajući sve ovo u vidu, svaki model podataka treba da čine sledeće osnovne komponente:
 1. **Struktura modela.**
 2. **Ograničenja** - semantička ograničenja na vrednosti podataka koja se ne mogu predstaviti samom strukturom modela.
 3. **Operacije** nad konceptima strukture, preko kojih je moguće prikazati i menjati vrednosti podataka u modelu;

MODEL PODATAKA

- Pri analizi različitih modela podataka analiziraće se način specifikacije i implementacije *strukture, ograničenja, operacija i dinamičkih pravila integriteta* za svaki konkretan model.
- Osnovni problem u modeliranju je složenost sistema, odnosno mnoštvo objekata, atributa objekata i veza između objekata, koje postoje u jednom realnom sistemu. Opšti metodološki pristup za savladavanje ove složenosti u opisu sistema je **apstrakcija**. **Apstrakcija** je kontrolisano i postepeno uključivanje detalja u opis sistema, "sakrivanje" detalja u opisivanju sistema, odnosno izvlačenje i prikazivanje opštih, a odlaganje opisivanja detaljnih osobina nekog sistema.
- Koje apstrakcije i kako pojedini modeli koriste?

MODEL PODATAKA

POSTOJI VIŠE KRITERIJUMA ZA KLASIFIKACIJU MODELA PODATAKA:

- NAČIN OPISIVANJE DINAMIKE SISTEMA
- NAČIN OSTVARIVANJA OSNOVNIH CILJEVA SUBP-a
- DA LI SE MODEL KORISTI SAMO ZA PROJEKTOVANJE BP, SAMO KAO OSNOVA ZA NEKI SUBP ILI I JEDNO I DRUGO
- NAČIN KAKO PRETSTAVLJAJU OBJEKTE I VEZE

MODEL PODATAKA

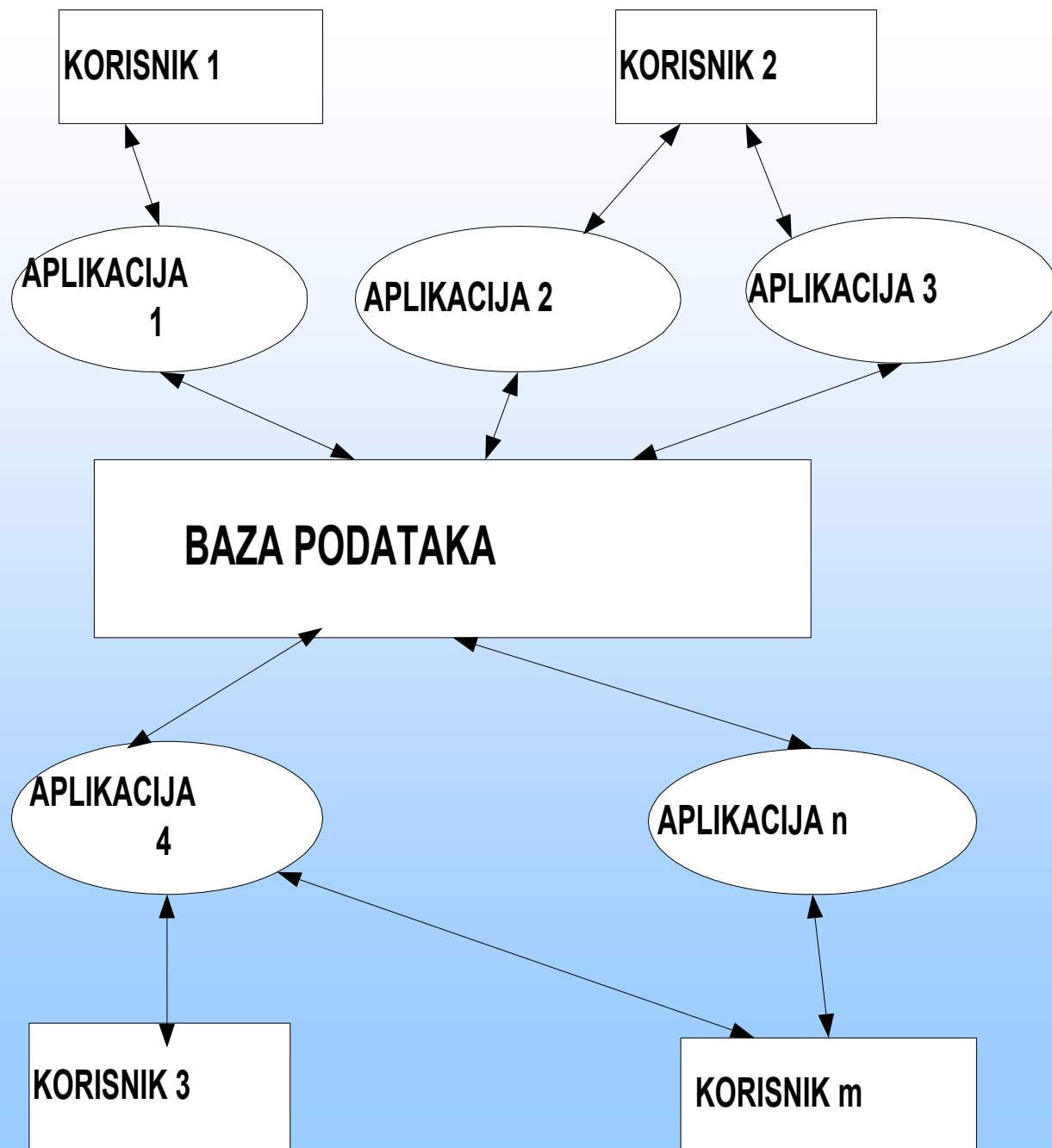
POREĐENJE - KRITERIJUM:
OPISIVANJE DINAMIKE SISTEMA
MOGUĆA PODELA:

- ❑ KONVENCIONALNI (HIJERARHIJSKI, MREZNI, RELACIONI, MODEL OBJEKTI-VEZE)
- ❑ OBJEKTNI
- ❑ AKTIVNE BAZE PODATAKA

KONVENCIONALNI:

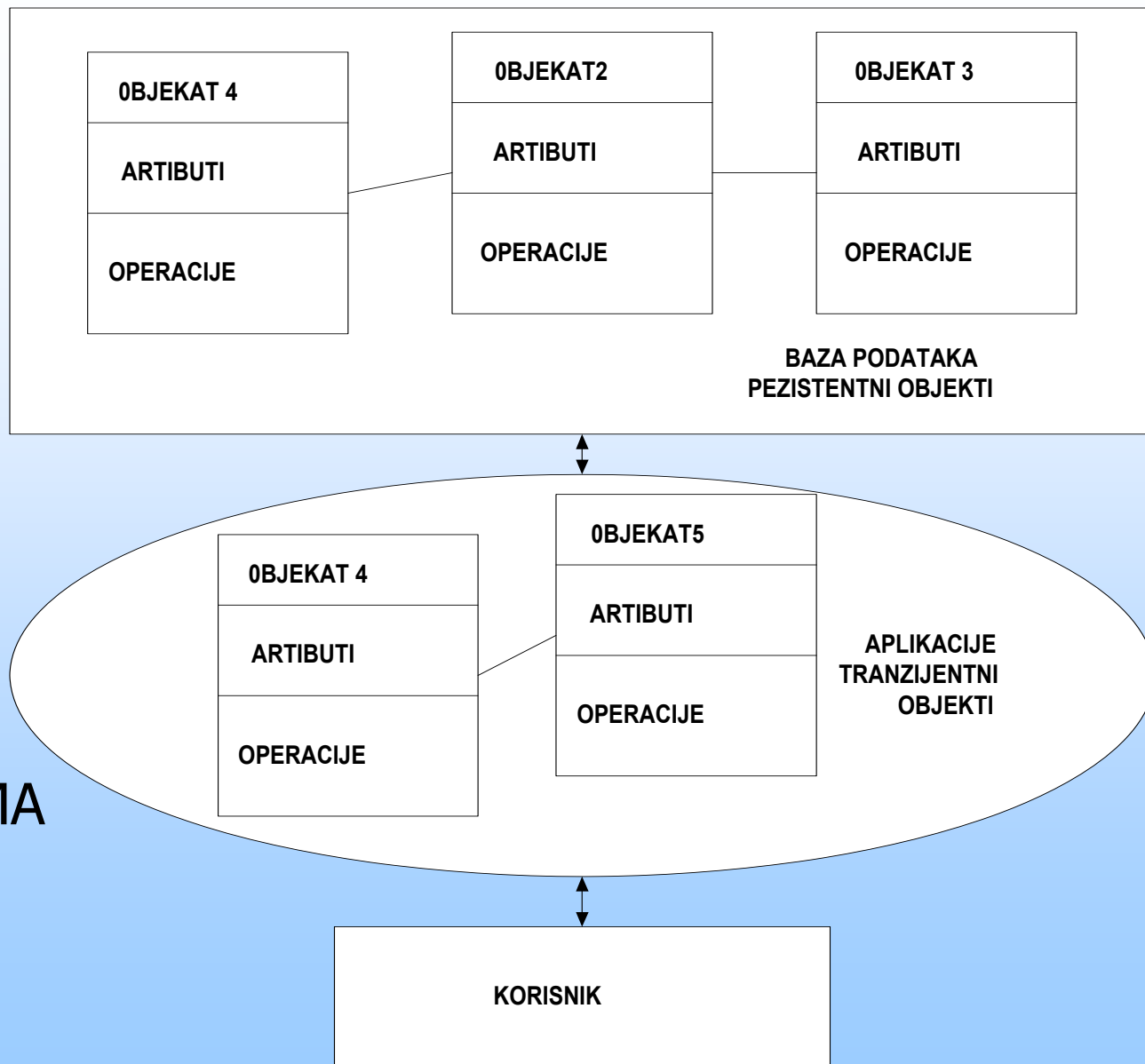
BAZA PODATAKA
JE POTPUNO
STATIČKI KONCEPT
SA IZUZETKOM
JEDNOSTAVNIH
DINAMIČKIH
PRAVILA
INTEGRITETA

SVA DINAMIKA JE
U APLIKACIJAMA



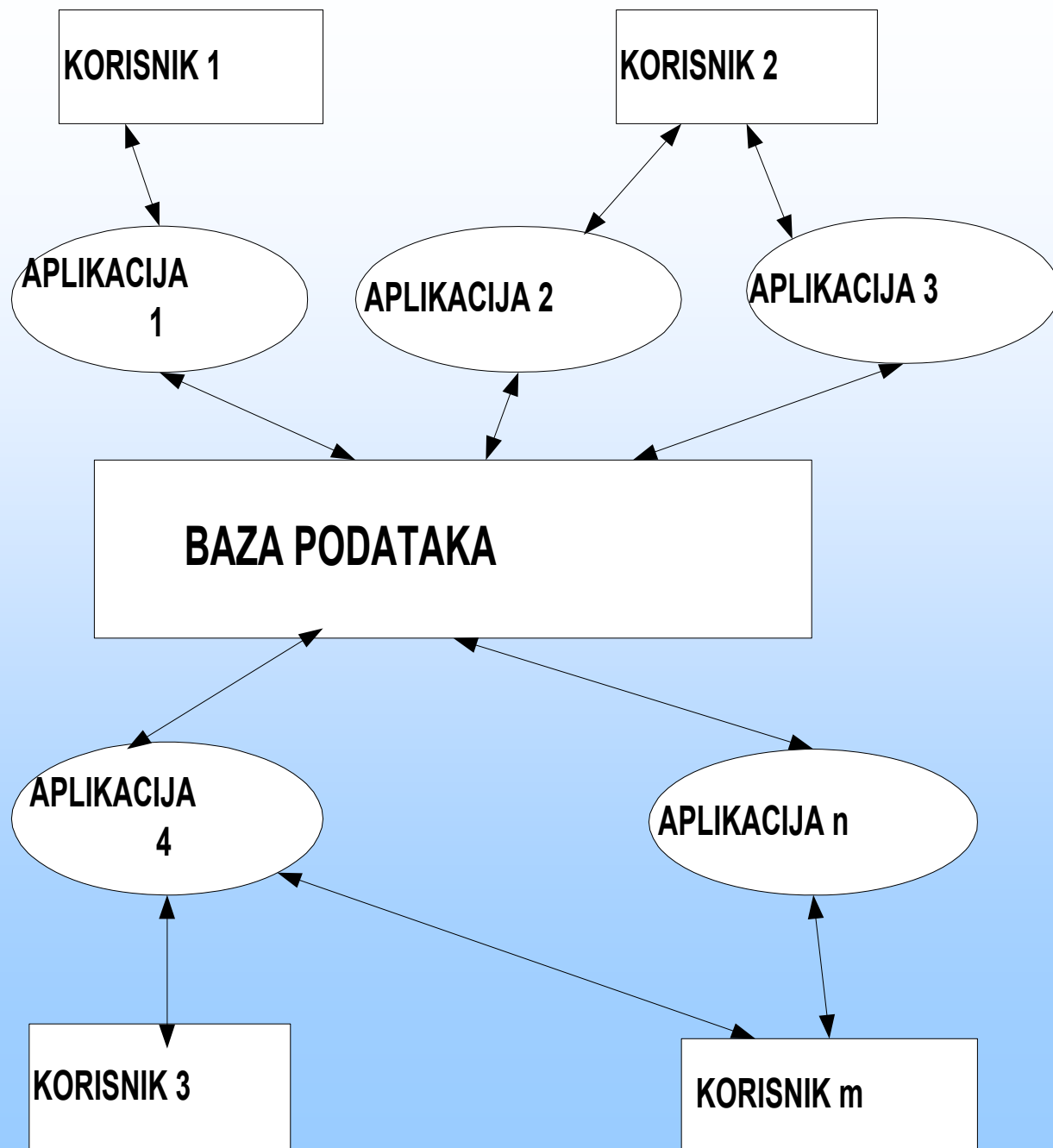
OBJEKTNI MODEL

DINAMIKA SE
OBUHVATA NA
ISTI NAČIN I U
BAZI PODATAKA
I U APLIKACIJAMA



AKTIVNE BAZE:

PREKO KONCEPTA
PRAVILA KOJA
SE ISKAZUJU KAO
KOMBINACIJA
<USLOV, AKCIJA>,
ZNATNO VEĆA
"KOLIČINA"
DINAMIKE SISTEMA
SE NALAZI U BAZI
PODATAKA



MODEL PODATAKA

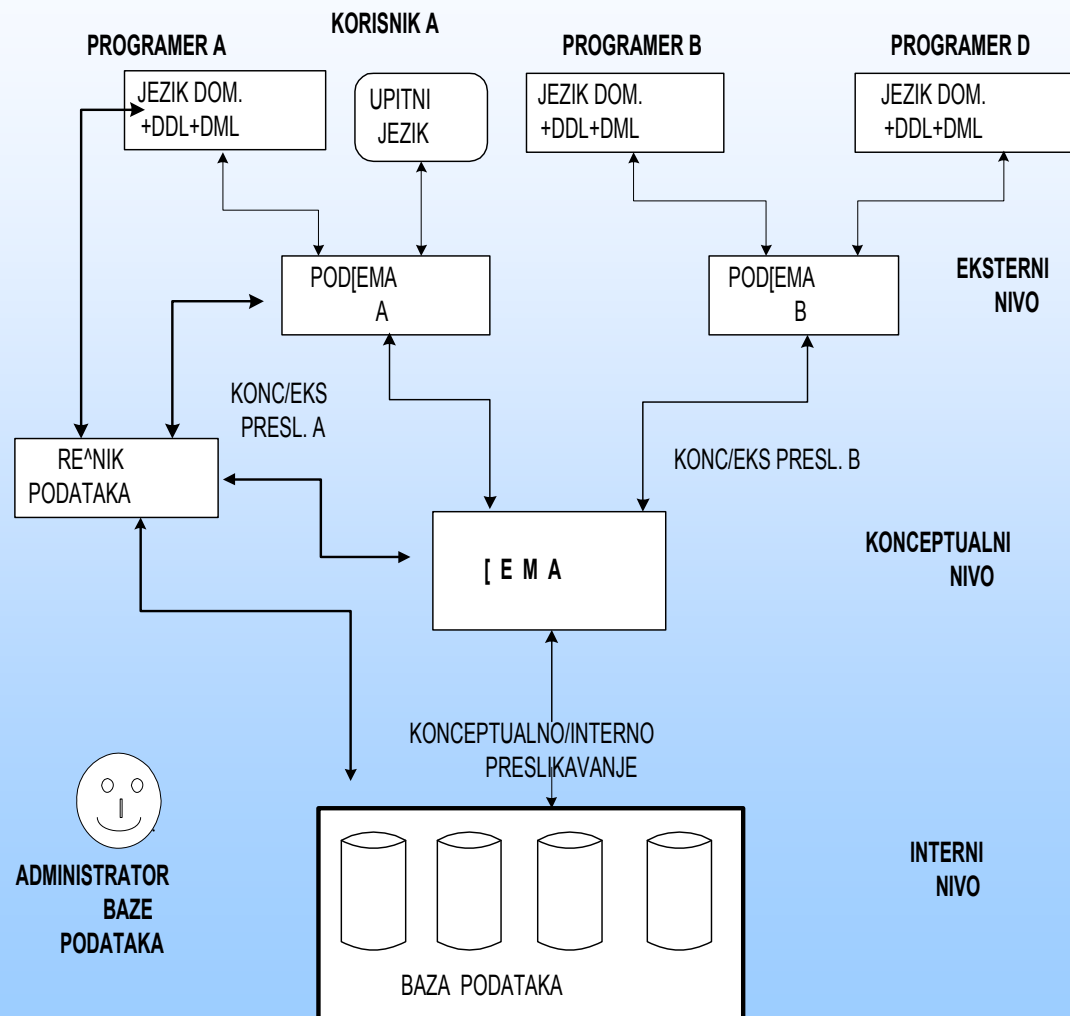
POREĐENJE - KRITERIJUM: OSTVARIVANJE OSNOVNIH CILJEVA SUBP-A

OSNOVNI CILJEVI BP:

- (1) Neredundatno pamćenje podataka
- (2) Višestruko paralelno (konkurentno) korišćenje podataka
- (3) Ostvarivanje nezavisnosti programa i logičke i fizičke strukture baze podataka)

- ☐ KONVENCIONALNE BP
- ☐ OBJEKTNE BP

KONVENCIONALNE BAZE- ANSI/SPARC STANDARDNA ARHITEKTURA



OSNOVNAI CILJEVI BP - OBJEKTNE BAZE

- Objektne baze prva dva cilja ispunjavaju na sličan način kao i konvencionalne: (1) Neredundantost podataka se ostvaruje dobrim projektovanjem BP, (2) Višestruko paralelno (konkurentno) korišćenje podataka, odnosno upravljanje obradom transakcija, ostvaruje se sličnim mehanizmima "zaključavanja"
- Nezavisnosti programa i logičke i fizičke strukture baze podataka ostvaruje se preko **koncepta nezavisnosti specifikacije tipa objekta od njegove implementacije.**

NEZAVISNOST PROGRAMA I STRUKTURE BAZE PODATAKA - SPECIFIKACIJA I IMPLEMENTACIJA OBJEKATA

- Objekti koji imaju isti skup stanja (isti skup atributa i veza) i isto ponašanje (isti skup operacija) mogu se predstaviti opštim **tipom objekta**.
- Svaki tip se može posmatrati na dva načina: kao specifikacija - **interfejs tipa** koji definiše "spoljne", vidljive karakteristike tipa i kao jedna ili više **klasa**. **Klasa** je specifikacija implementacije tipa i pretstaljena je strukturom podataka i skupom metoda koje predstavljaju implementaciju operacija definisanih u interfejsu tipa. Jedan tip može imati više različitih klasa kao svoje implementacije.

NEZAVISNOST PROGRAMA I STRUKTURE BAZE PODATAKA - SPECIFIKACIJA I IMPLEMENTACIJA OBJEKATA

- Interfejsi tipova objekata, odnosno sve vidljive karakteristike tipova objekata baze podataka čine konceptualni model baze podataka.
- Još je pogodnije objektnu bazu podataka posmatrati kao jedinstvenu softversku komponentu.
- KOMPONENTA je fizički, izmenljivi deo sistema, realizacija skupa interfejsa

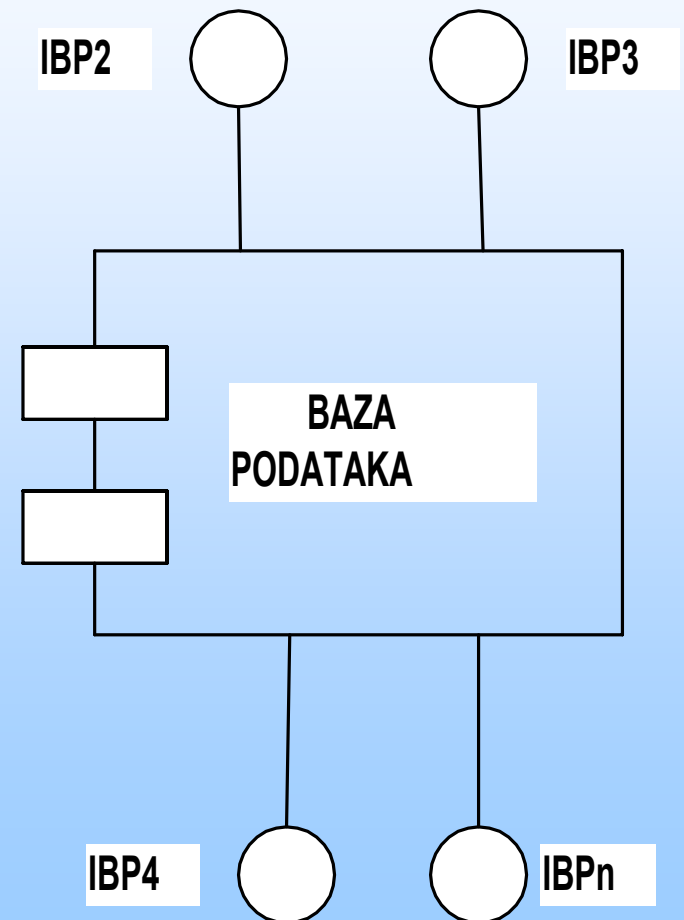
NEZAVISNOST PROGRAMA I STRUKTURE BAZE PODATAKA

- SPECIFIKACIJA I IMPLEMENTACIJA OBJEKATA

KOMPONENTA - FIZIČKA BAZA
PODATAKA

**UNIJA SVIH INTERFEJSA -
KONCEPTUALNI MODEL**

**SPECIFIČNI INTERFEJSI -
KORISNIČKI PODMODELI**



MODEL PODATAKA

- ZA PROJEKTOVANJE: MODEL OBJEKTI VEZE, OBJEKTNI MODEL, RELACIONI MODEL
- KAO OSNOVA SUBP-a (IMPLEMENTACIJA): HIJERARHIJSKI, MREŽNI, RELACIONI, OBJEKTNI

*Najčešća kombinacija za razvoj softvera danas:
Objektni pristup i jezici za razvoj aplikacija i relacioni
SUBP.*

MODEL PODATAKA

POREĐENJE - KRITERIJUM:

NAČIN KAKO PRETSTAVLJAJU OBJEKTE I VEZE:

- ❑ VREDNOSNO ORJENTISANI: vrednosti atributa se koriste i za identifikaciju objekata i za pretstavljjanje veza: Relacioni model
- ❑ OJEKTNO ORJENTISANI: Objekti se identifikuju prilikom kreiranja, veze se uspostavljaju preko "pokazivača": Hijerarhijski, Mrežni, Objektni,

MODEL PODATAKA

	Proste strukture	Složene strukture
Upitni jezik	Proste strukture i upitni jezik (konvencionalni relacioni SUBP)	Složene strukture i upitni jezik (Relaciono- objektni SUBP)
Bez upitnog jezika	Jednostavne strukture bez upitnog jezika (Mrežni i hijerarhijski model)	Složene strukture bez upitnog jezika (Objektno-orjentisani modeli)

JEDAN MOGUĆI PRISTUP POREĐENJU
RAZLIČITIH MODELA

MODEL PODATAKA -STANDARDI

- MREŽNI MODEL -CODASYL STANDARD
- KONVENCIONALNI RELACIONI MODEL - STANDARD SQL2 (SQL92)
- OBJEKTNI MODEL - ODMG 2.0 (3.0 APRIL 2000.) STANDARD
- OBJEKTNO-RELACIONE BAZE (STANDARD SQL3-SQL:1999)

SADRŽAJ KURSA (1/2)

■ I DEO : MODEL PODATAKA

1. Model objekti-veze
2. Relacioni model
3. Standardni upitni jezik SQL
4. Objektne baze podataka
5. Objektno-relacioni model
6. Aktivne baze podataka
7. XML kao model podataka

SADRŽAJ KURSA (2/2)

II DEO: FUNKCIJE SISTEMA ZA UPRAVLJANJE BAZOM PODATAKA

1. Fizička struktura baze podataka
2. Optimizacija upita
3. Upravljanje izvršenjem transakcija i oporavak baze podataka
4. Sigurnost baze podataka
5. Katalog baze podataka
6. Distribuirane arhitekture

III DEO: PROJEKTOVANJE BAZA PODATAKA

1. Analiza sistema i zahteva korisnika: SSA
2. Analiza sistema i specifikacija aplikacija: objektne metode
3. Konceptualno modelovanje
4. Logičko i fizičko projektovanje baza podataka

IV DEO-BAZE PODATAKA I RAZVOJ APLIKACIJA

1. SQL unutar klasičnih i objektnih programskih jezika (embedded SQL)
2. Pristup bazama podataka preko poziva funkcija
3. Procedure baze podataka - Persistent Stored Module (PSM)

REFERENCE (1/5): NAJNOVIJE

1. C.J. Date: *An Introduction to Database Systems*, 7th edition, Addison Weseley, 2000.
2. H.G.Molina, J.Ullman, J Widom, *Database Systems, The Comlete Book*, Prentice Hall, 2002
3. J.Ullman, J Widom, *A First Course in Database Systems*, Prentice Hall, 2002
4. R.A.Elmasri, B.S.Navate, *Fundamentals of Database Systems*, 3rd edition (August 1999) Addison-Wesley Pub Co
5. P.Atzeni, S.Ceri, S.Parabichi. R.Torlone, *Database Systems*, McGrawHill, 1999

REFERENCE (2/5): STARE DOBRE

1. C.J.Date: *An Introduction to Databsase Systems*, 6th edition , Addison-Wesley,1995
2. J.Ullman, J.Widom: *A First Course in Database Systems*, Prentice Hall, 1997
3. J.D.Ullman: *Principles od Database and Konwledge-Base Systems*, Vol I, II, Computer Science Press, 1988.
4. B. Lazarević, *Baze podataka, materijal za studente*, Fon 1992.

REFERENCE (3/5)

RELACIONI MODEL – TEORIJSKE REF.

1. E.F. Codd, *The Relational model for Database management*, , Version 2, Addison-Wesley, 1990.
2. D. Maier, *The Theory of Relational Databases*, Computer Science Press, 1983.
3. Chao-Chih Yang, *Relational Databases*, Prentice Hall, 1986
4. S. Alagić, *Relacione baze podataka*, Svjetlost, Sarajevo 1984.
5. P. Mogin, I. Luković, *Principi baza podataka*, Stylos, Novi Sad, 1996
6. P. Mogin, I. Luković, M. Govedarica, *Principi projektovanja baza podataka*, Univerzitet u Novom Sadu, 2000.

REFERENCE (4/5)

OBJEKTNE BAZE - OSNOVNE REFERENCE

1. R.C.G.Cattell, *The Object Database Standard: ODMG-93*, Release 1.2., Morgan Kaufman Publishers, 1996.
2. R.C.G.Cattell, D.K.Barry, *The Object Database Standard: ODMG2.0.*, Morgan Kaufman Publishers, 1997.

Sve novije knjige o bazama podataka

REFERENCE (5/5)

OBJEKTNE I OBJEKTNO-RELACIONE BAZE PODATAKA

1. M. Loomis, *ObjectDatabases: the Essential*, Addison Weseley, 1995.
2. M.Stonebraker *Object-Relational Databases*, Morgan Kaufmann Publ., 1996.
3. SQL3 Standard, <ftp://jery.ece.umassd.edu/SCG32/WG3/>
4. Sve novije knjige o bazama podataka