

# Modeli podataka

---

- Osnovni pojmovi modela podataka
- Primeri MOV-a
- Logi ko modeliranje podataka (6 koraka)
- Tipovi veza kod IDEF1X metodologije
- Logi ko modeliranja podataka na primeru fakture

## Model podataka - osnovne komponente

---

- ❑ Podatak je kodirana injenica iz realnog sistema, on je nosilac informacije.
- ❑ Informacija je protuma eni (interpretirani) podatak.
- ❑ Interpretacija podataka se vrši na osnovu strukture podataka, semanti kih ograni enja na njihove vrednosti i preko operacija koje se nad njima mogu izvršiti.

## Model podataka - osnovne komponente

- (1) **Struktura modela** - objekti, atributi, veze
- (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;

## Modeliranje podataka

- **Modeliranje podataka** definiše se kao :
  - **Model Objekti-Veze-MOV ili E-R (Entity-Relationship) model**
  - tehnika za organizovanje i dokumentovanje strukture podataka
  - pojednostavljeno predstavljanje realnog sistema preko *skupa objekata (entiteta), veza između objekata i njihovih atributa.*
  - stanje sistema u jednom trenutku vremena, predstavljeno preko skupa podataka i njihovih međusobnih veza,
  - predviđanje budućih izlaza sistema na bazi poznavanja ulaza
- **Notacije modela objekti-veze:**
  - imenovani prema njihovim tvorcima:
    - *Chen, Martin, Bachman, Merise i dr., i*
    - **prema objavljenim standardima, npr., IDEF1X.**
  - svi „jezici“ modeliranja podataka podržavaju iste fundamentalne koncepte i konstrukcije.

## Modeliranje podataka

- **I korak: određivanje osnovnih objekata sistema**
  - (tehnika intervju-a- pitati vlasnika i korisnika sistema npr. o kupcima i proizvodima)
- **II korak: identifikovanje atributa**
  - (mnogi atributi imaju istu osnovu: npr. ime adresu, NazivKupca, NazivKlijenta)
- **III korak: identifikacija ključeva za svaki objekat**
  - (vrednost ključeva ne sme da se menja – n.pr. prezime osobe; vrednost ključeva ne može da bude prazno polje, kontrole ključeva moraju biti instalirane)
- **IV korak: povezivanje objekata**
  - (objekat od koga se uspostavlja veza zove se “roditelj” ili domen, a objekat ka kome je uspostavljena veza zove se “dete” ili ko-domen. Logički model se definiše: strukturom podataka, ograničenjima –poslovna pravila, struktura operacija – održavanje baze i pretraživanje)
- **V korak: normalizacija relacionog modela**
  - (tehnika analize – da li su atributi dobro postavljeni...)

**Poslednji korak je prevođenje logičkog modela podataka u fizički model podataka (šema baze podataka)**

## Osnovni pojmovi modela podataka

- **Objekat:**
  - element *klasa osoba, mesta, fizičkih objekata, događaja ili koncepata* o kojima prikupljamo i skladištimo podatke.
  - nešto što se može videti, dodirnuti
  - ima svoja svojstva i ponašanja i o kome korisnici mogu da skladište podatke
- **Tipovi objekata** klasifikuju se u:
  - *osobe* - radnici, klijenti, prodavci, studenti i dr.
  - *mesta* - skladišta, zgrade, sobe i dr.
  - *stvari* - proizvod, vozilo, opremu, videotraku i dr.
  - *događaji* - porudžbina, planiranje, račun, aplikacija, registraciju ili rezervacija.

- **Atribut:**
  - osobina ili karakteristika objekta.
  - preslikavanje iz **klase** objekta u **domen** (MLB: **Radnik**-----> **Matbr**)
- **Tip podatka:**
  - definiše klasu podataka koja se može skladištiti
- **Domen atributa:**
  - definiše koje vrednosti može da ima jedan atribut
- **Difoltna vrednost:**
  - vrednost koja se bitno uskladištena za dati atribut ukoliko je korisnik ne promeni

- **Jedinstveni ključ :**
  - osnovna svrha ključa jeste da jedinstveno identifikuje svaki objekat,
  - svaki objekat mora da ima *jedinstveni ključ* po kome se pretražuje baza podataka.
- **Složeni ključ - grupa atributa koja jedinstveno identifikuje objekat.**
  - ŠIFRA\_KASETE (PRIMARY KEY)
  - ŠIFRA\_NASLOVA
  - BROJ\_KOPIJE (Složeni ključ : **ŠIFRA\_NASLOVA, BROJ\_KOPIJE**)
- **Kandidati za ključ :**
  - kandidati za primarni ključ ,
  - objekat može imati **više od jednog** ključa.

**Primer:** objekat RADNIK se može jedinstveno identifikovati preko: *matičnog broja ili šifre zaposlenog ili e-mail adrese*. Svaki od ovih atributa se nazivaju kandidati za ključ .
- **Primarni ključ (PRIMARY KEY):**
  - kandidat za ključ koji se najčešće koristi da jedinstveno identifikuje dati objekat.
- **Alternativni ključevi:**
  - svi drugi kandidati za ključ koji nisu izabrani za primarni ključ .
- **Difoltna vrednost primarnog ključa je NOT NULL:**
  - ključ ne sme da bude prazno polje, jer *onda* ne može i da jedinstveno identifikuje dati objekat.

## Osnovni pojmovi modela podataka

- **Relacije objekata:**
  - objekti ne egzistiraju sami ve moraju biti u nekoj **relaciji** ili **vezi** sa drugim objektima.
- **Asocijacija:**
  - predstavlja istovremeno i objekat i vezu, odnosno udruženi objekat (*associative entity*), od dva ili više objekta.
- **Kardinalnost:**
  - definiše preslikavanje E1 → E2 i E2 → E1 isti e ulogu objekta i veze
  - definiše **minimalni** i **maksimalni** broj doga aja jednog objekta koji se nalazi u konkretnoj relaciji sa drugim objektom (1:1, 1:N, N:1, N:N)
  - pošto su sve relacije dvosmerne, kardinalnost se mora definisati za oba smera.
  - **1. preslikavanje** - semantika: **radnik radi u odeljenju**
  - **2. preslikavanja** – semantika: **odeljenje zapošljava radnika**



6.11.2015

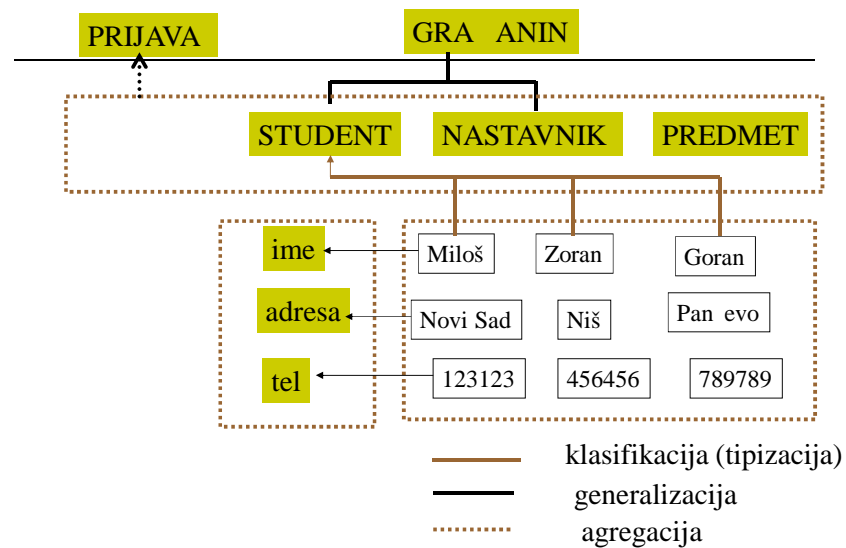
Poslovni informacioni sistemi

9

## APSTRAKCIJE U MODELU PODATAKA

- KLASIFIKACIJA (tipizacija)
- GENERALIZACIJA
- AGREGACIJA

## PRIMER APSTRAKCIJA

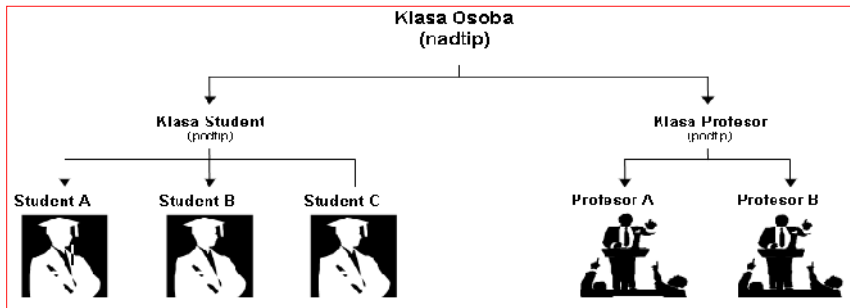


## Vrste modela

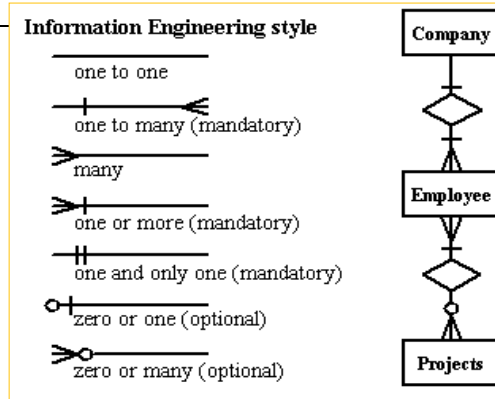
- Model Objekti - Veze
- Relacioni model
  
- *Istorija*
  - Hijerarhijski model
  - Mrežni (CODASYL) model

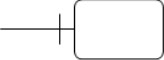
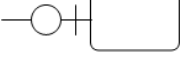
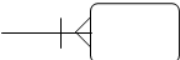


Osnovni pojmovi modela podataka

- **Generalizacija (apstrakcija):**
  - tehnika gde se objekti sa zajedni kim (sli nim) atributima, vezama i/ili operacijama, grupišu (generalizuju) u jedan objekat koji se zove *nadtip*.
- **Specijalizacija:**
  - inverzni postupak, gde se za neki tip objekta, definišu njegovi *podtipovi*, koji imaju neke njima specifi ne attribute, veze i/ili operacije.



Relacija - Kardinalnost entiteta  
precizira koliko instanci jednog entiteta se odnosi na drugi entitet



| Interpretacija kardinalnosti | Minimum | Maksimum | Grafička notacija   |
|------------------------------|---------|----------|---|
| Tačno jedan                  | 1       | 1        |   |
| Nula ili jedan               | 0       | 1        |   |
| Jedan ili više               | 1       | >1       |   |
| Nula, jedan ili više         | 0       | >1       |   |
| Više od jedan                | >1      | >1       |  |

## Model Objekti-Veze (MOV)

### *E-R Model, Entity-Relationship Model*

- **Objekat (entitet)** - grupa (skup) sadržaja sa karakteristikama koje su bitne za celinu.
- **kandidati za objekte:**
  - Fizi ki objekti (vozila, mašine,...)
  - Osobe
  - Lokacije (mesta, adrese, koordinate...)
  - Organizacije
  - Grupe/klase/tipovi (proizvoda, poslova...)
  - Dokumenta
  - Pridruženja (zadatak-osoba, vozilo-vožnja)
  - Pripadnost/ lanstvo

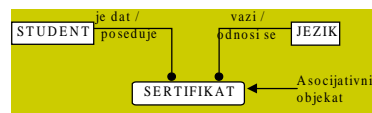


## MOV – vrste objekata

- **Nezavisan objekat** ima osobinu koja ga može jednoznačno identifikovati (ne zavisi od drugih objekata).
- **Zavisan objekat** je onaj čija egzistencija i identifikacija zavise od drugog (ili drugih) objekata.

Postoje

- *karakterističan objekat (slab objekat)* – onaj koji se ponavlja više puta za određeni nezavisni objekat;
- *asocijativni objekat*, koji predstavlja vezu više objekata;

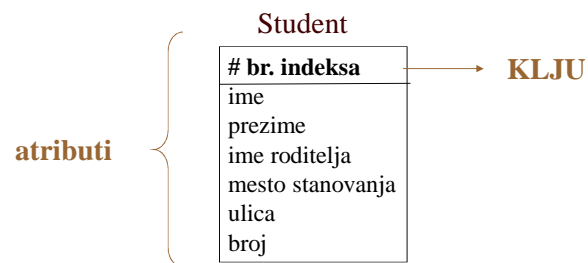


## MOV - atributi

- **Atributi** su karakteristike ili osobine iskazane kao jedna ili više vrednosti koje opisuju objekat. Svaki atribut ima svoje ime.

## Ključ objekta




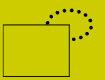
- Ključ je vrsta atributa koji jedinstveno identifikuje svaki primerak objekta.
- Od atributa - kandidata za ključeve bira se jedan koji postaje **primarni ključ**.
- Nijedan deo primarnog ključa ne može biti prazan ili nedostajati.



## Ključevi

- Ako ključ ima samo jedan atribut, onda je **prost ključ**; u suprotnom je **složen**.
- **Alternativni ključ** predstavlja atribut ili grupa atributa koji jedinstveno identifikuju primerke entiteta, ali postoje objekti za koje taj atribut nije definisan
- **Preneseni ključ** (Foreign Key) je atribut koji povezuje objekat 'dete' sa objektom 'roditelj'

## Veze (Relationship)

- identifikuju e (dete - roditelj) 
- neidentifikuju e
  - obavezne 
  - neobavezne 
- rekurzivne (na sebe samog) 
  - primer : radnik - šef


## Kardinalnost veza

### Kardinalnost veza roditelj - dete

- nula, jedan ili više
- jedan ili više
- nula ili jedan
- ta no n (*primer : godišnja doba, posada aviona*)

### Kardinalnost veza dete - roditelj

- dozvoljena nula
- nije dozvoljena nula

Veza "više prema više" 

## Referencijalni integritet

---

- Omogućava korektno povezivanje objekata
- Definiše se za svaku vezu, posebno za roditelja, posebno za dete
- Dolazi do izražaja kod održavanja modela
- Realizuje se putem ograničenja, operacija i akcija

## Ograničenja

---

- Nad strukturom
  - Integritet entiteta
  - Nad standardnim domenom
  - Tip, dužina podataka
- Nad vrednošću u domenu
  - Dozvoljene vrednosti
- Na kardinalnost
  - (0,1,n), (1,n), (0,1), (Exactly)

## Operacije

---

- nad konceptima strukture, po ograničenjima
  - INSERT (ubacivanje)
  - REPLACE (ključ, deo ključa)
  - DELETE (objekat, veza, roditelj)

## Akcije

---

- Restrict - odbija operaciju koja narušava integritet
- Cascade - prosleđuje operaciju
- Default - kreira pretpostavljeni objekat
- Set null - nepoznato pojavljivanje
- None - nema ograničenja

## Postupak normalizacije

“jedna činjenica na jednom mestu”

- uklanjanje redundanse\*!

- Prva normalna forma
  - Svaki od atributa ima jedno značenje i ne više od jedne vrednosti za svaki primerak (instancu)
- Druga normalna forma
  - Svaki atribut koji nije ključ potpuno zavisi od primarnog ključa
- Treća normalna forma
  - Svaki atribut koji nije ključ mora da zavisi jedino od primarnog ključa

\* redundansa – višestruko ponavljanje istog podatka u bazi

## 1NF - primer

### RADNIK

|                             |
|-----------------------------|
| Šifra radnika               |
| Prezime                     |
| Ime                         |
| Kvalifikacija               |
| Dat.zaposl. ili dat.odlaska |

|     |            |            |            |
|-----|------------|------------|------------|
| 123 | Petar Peri | Programer  | 01.11.1998 |
| 124 | Ana Ili    | Projektant | 20.05.2005 |
| 125 | Milan Mili | Operater   | 15.09.2004 |

1NF : Jednoznačna upotreba atributa;  
Svaki od atributa ima jedno značenje  
i ne više od jedne vrednosti za svaki primerak (instancu)

## 2NF - Primer

### ISPLATA

Šifra radnika  
Br. isplate  
~~Datum zapoclenja~~  
Isplata

|     |   |            |       |
|-----|---|------------|-------|
| 123 | 1 | 01.10.1997 | 20000 |
| 123 | 2 | 01.10.1997 | 22500 |
| 123 | 3 | 01.10.1997 | 22000 |
| 123 | 4 | 01.10.1997 | 25000 |

2NF : Svaki atribut koji nije ključ mora potpuno da zavisi od primarnog ključa, inače ga treba premestiti u nadređeni entitet

## 3NF - Primer

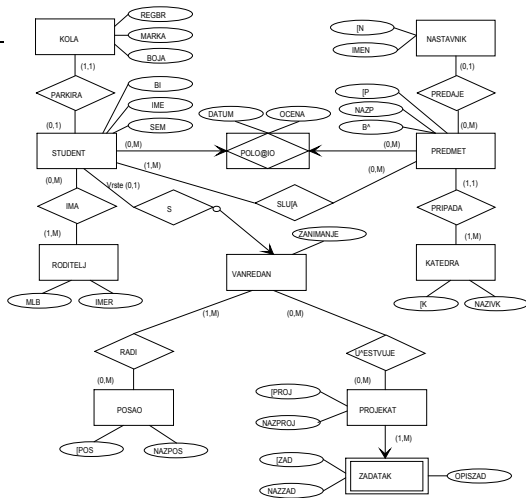
### Radna lista

Broj liste  
Br. asova  
Cena asa  
~~Iznos~~

|     |    |     |      |
|-----|----|-----|------|
| 321 | 10 | 100 | 1000 |
| 322 | 2  | 80  | 160  |
| 323 | 3  | 150 | 450  |

3NF : Svaki atribut koji nije ključ mora da zavisi jedino od primarnog ključa; Ne koristiti attribute čija se vrednost može izračunati

## Osnovni koncepti MOV



## Osnovni pojmovi modela podataka

- **Objekat** je klasa osoba, mesta, objekata, događaja ili koncepata o kojima treba da prikupljamo i skladištimo podatke.
- **Objekat** je nešto što se može videti, dodirnuti ili drugačije osetiti, koji ima svoja svojstva i ponašanja i o kome korisnici mogu da skladište podatke.
- Tipovi objekata se mogu klasifikovati u osobe, mesta, stvari ili događaje. U okviru tipa objekta *osobe* mogu se svrstati radnici, klijenti, prodavci, studenti i dr. Skladišta, zgrade, sobe su primeri tipa objekata *mesta*. Primeri tipa objekata *stvari* uključuju proizvod, vozilo, opremu, videotraku i dr. Na kraju objekti *događaja* uključuju porudžbinu, planiranje, aplikaciju, registraciju ili rezervaciju.
- **Atribut** je osobina ili karakteristika objekta.
- **Tip podatka** definiše koja klasa podataka može biti skladištena u taj atribut.
- **Domen** definiše koje vrednosti može da ima jedan atribut.
- **Difoltna vrednost** je ona vrednost koja može biti uskladištena za dati atribut ukoliko je korisnik ne promeni.



## Osnovni pojmovi modela podataka (*nastavak*)

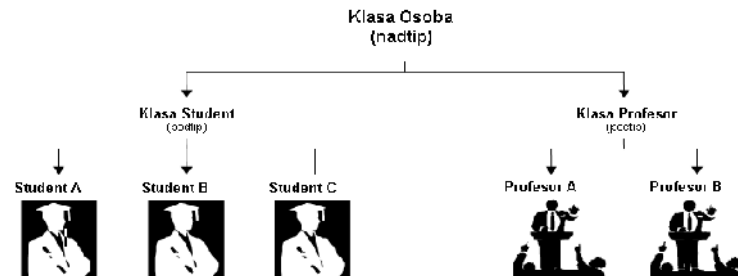
- Svaki objekat mora da ima jedinstveni **klju** po kome e se pretraživati u bazi podataka. Osnovna svrha klju a jeste da jedinstveno identifikuje svaki objekat.
- Grupa atributa koja jedinstveno identifikuju objekat se zovu **složeni klju** .  
ŠIFRA\_KASETE (PRIMARY KEY)  
. ŠIFRA\_NASLOVA  
. BROJ\_KOPIJE
- Objekat može imati više od jednog klju a. Na primer, objekat RADNIK se može jedinstveno identifikovati preko matičnog broja ili preko šifre zaposlenog ili preko e-mail adrese. Svaki od ovih atributa se nazivaju kandidati za klju . **Kandidati za klju** su kandidati za primarni klju .
- **Primarni klju** (PRIMARY KEY) je kandidat za klju koji e se naj če e koristiti da jedinstveno identifikuje dati objekat. Svi drugi kandidati za klju koji nisu izabrani za primarni klju se zovu **alternativni klju evi**.
- Dofoltna vrednost primarnog klju a je NOT NULL, odnosno klju ne sme da bude prazno polje, jer onda ne e mo i da jedinstveno identifikuje dati objekat.

## Osnovni pojmovi modela podataka (*nastavak*)

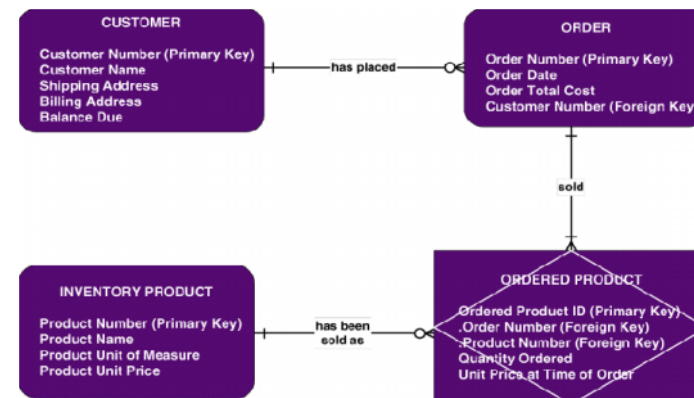
- Objekti ne ekzistiraju sami ve moraju biti u nekoj **relaciji** ili **vezi** sa drugim objektima.
- **Asocijacija** istovremeno predstavlja i objekat i vezu, odnosno udruženi objekat (*associative entity*), izme u dva ili više objekta.
- **Kardinalnost** definiše minimalni i maksimalni broj doga aja jednog objekta koji se nalazi u konkretnoj relaciji sa drugim objektom. Pošto su sve relacije dvosmerne, kardinalnost se mora definisati za oba smera.
- Postoje nekoliko notacija modela podataka. Neka su imenovana po njihovim pronalaza ima, kao što su Chen, Martin, Bachman, Merise ili po objavljenom standardu IDEF1X, UML i dr.

## Osnovni pojmovi modela podataka (nastavak)

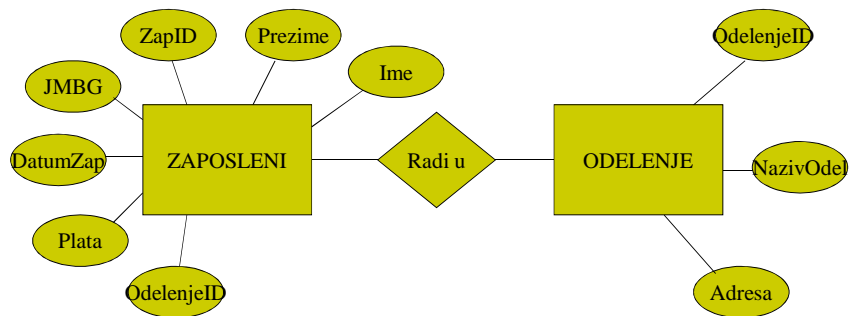
- **Generalizacija** je tehnika gde se objekti sa zajedni kim atributima, vezama i/ili operacijama, grupišu (generalizuju) u jedan objekat koji se zove **nadtip**. Inverzni postupak, gde se za neki tip objekta, definišu njegovi podtipovi, koji imaju neke njima specifi ne atribute, veze i/ili operacije, je **specijalizacija**.



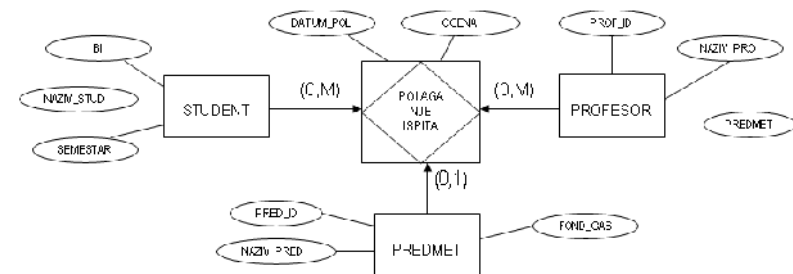
## Jednostavan primer modela objekti-veze (MOV)



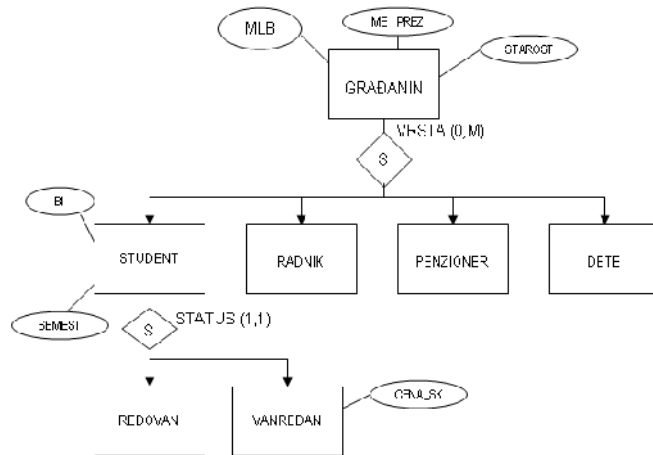
### Relacija zaposleni-odelenje



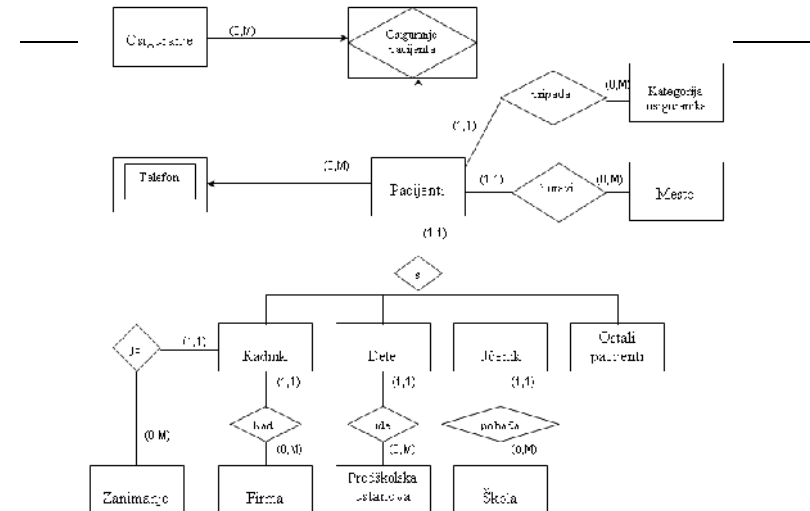
### Agregacija "polaganje ispita"



## Generalizacija/specijalizacija



## Dijagram osiguranja pacijenata





## Logi ko modeliranje podataka (6 koraka)

1. Prvi zadatak u modeliranju podataka je da se odrede **osnovni objekti sistema**. Postoji nekoliko tehnika koje se mogu koristiti za identifikaciju objekata:
  - Tokom intervjua ili JAD (*Joint Application Development*) sesije, sa vlasnicima i korisnicima sistema, treba obratiti pažnju na ključne reči i njihove diskusije. Na primer, ukoliko korisnik kaže «Treba informisati naše stalne kupce, o novom kvalitetu proizvoda», primeti ćemo da su ključne reči u ovoj rečenici KUPCI i PROIZVOD, a oni su ujedno i objekti sistema.
  - U toku intervjua i JAD sesija, treba pitati vlasnike i korisnike sistema da identifikuju one stvari za koje žele da prikupljaju, skladište i dobijaju informacije.
  - Druga tehnika za identifikaciju objekta je da se prouče postojeći formulari i kartoteke. Neki formulari identifikuju događaje objekata. Na primer, porudžbina, uplata, depozit itd.
  - Neki CASE alati takođe mogu da identifikuju objekte.

<sup>1</sup>JAD – *Joint Application Development* (Zajednički razvoj aplikacija) je metod gde se na organizovanim grupnim sastancima prikupljaju činjenice, izgrađuju modeli podataka i izvršava simultana provera ovih modela.

## Logi ko modeliranje podataka (*nastavak*)

2. Drugi zadatak u modeliranju podataka jeste da se napravi **kontekstualni model podataka** koji sadrži samo osnovne, nezavisne objekte.
3. Treći zadatak je da se identifikuju **ključevi za svaki objekat**. Predlaže se sledeće:
  - Vrednost ključa ne sme da se menja u toku veka trajanja svakog objekta. Na primer, naziv osobe se ne može uzeti kao ključ jer osoba može da promeni svoje prezime ukoliko se venčala ili razvede.

Istovremeno treba da se identifikuje i **generalizacija**, odnosno nadtipovi i podtipovi strukture.
4. Nakon identifikacije ključeva objekta, treba da se odrede i ostali **atributi objekta**. Predlaže se sledeći i na njih:
  - Mnoge kompanije koriste standardna imena i skraćenice. Administrator podataka obično održava takve standarde.
  - Pažljivo birajte imena atributa. Mnogi nazivi atributa imaju istu bazu, npr ime, adresa, datum, njih bi trebalo razdvojiti.

## Logi ko modeliranje podataka (*nastavak*)

5. **Identifikovanje zavisnih entiteta** - entiteti ija egzistencija i identifikacija zavisi od drugog ili drugih entiteta.
6. **Definisane veze** - Veza je asocijacija izme u dva ili više entiteta, tj. predstavlja odnos koji postoji me u objektima, bilo u realnosti ili u mislima. Entitet od koga je uspostavljena veza zove se "roditelj" (*parent*) ili domen, a entitet ka kome je uspostavljena veza zove se "dete" (*child*) ili kodomen.

Na in povezivanja dva entiteta (*kardinalnost*) *pokazuje "koliko ne ega" od dva entiteta može biti uklju eno (sadržano).*

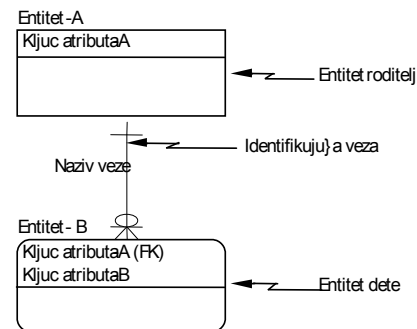
- Tipovi veza kod IDEF1X metodologije su:
  - Identifikuju e veze i
  - Neidentifikuju e veze:
    - obavezna neidentifikuju a veza i
    - neobavezna (opciona) neidentifikuju a veza.

## Martin-ova notacija kardinalnosti

| Interpretacija kardinalnosti | Minimum | Maksimum | Grafička notacija |
|------------------------------|---------|----------|-------------------|
| Tačno jedan                  | 1       | 1        |                   |
| Nula ili jedan               | 0       | 1        |                   |
| Jedan ili više               | 1       | >1       |                   |
| Nula, jedan ili više         | 0       | >1       |                   |
| Više od jedan                | >1      | >1       |                   |

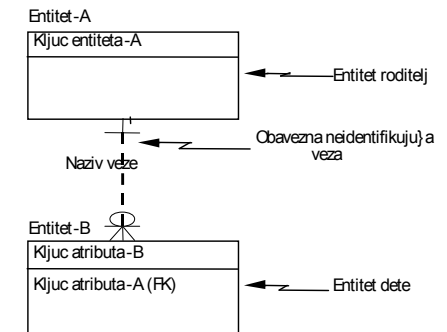
## Identifikuju e veze

- *Identifikuju e veze* - Veza se zove identifikuju a zato što klju evi entiteta "roditelj" predstavljaju deo identiteta entiteta "dete", tj. entitet "dete" zavisi od entiteta "roditelj" preko identifikatora.
- Prikazuje se punom linijom.
- U identifikuju oj vezi entitet "roditelj" ima svoj nezavisni primarni klju (Klju entiteta A), a entitet "dete" ima složeni klju koji se sastoji od svog klju a (Klju entiteta B) i prenesenog roditeljskog klju a (Klju entiteta A(FK)). Dakle, instance entiteta "roditelj" se definišu nezavisno, a instance entiteta "dete" se ne mogu identifikovati bez identifikatora entiteta "roditelj".



## Neidentifikuju e veze

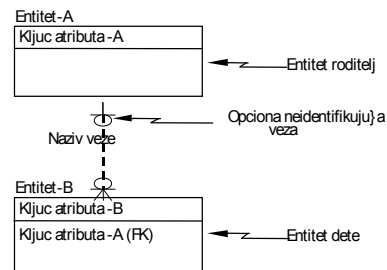
- Ako se svaki primerak entiteta "dete" može jedinstveno identifikovati, bez znanja veze sa primerkom entiteta "roditelj", onda se takva veza definiše kao *neidentifikuju a veza*.
- *Neidentifikuju e veze* su prikazane isprekidanom linijom koja povezuje entitet "roditelj" i entitet "dete" sa ta kom na strani entiteta "dete".





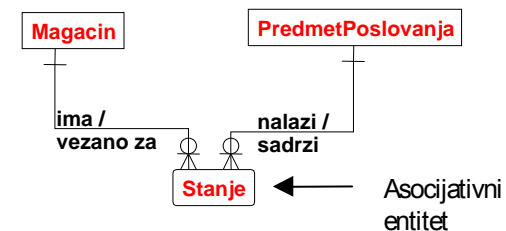
## Neidentifikuju a neobavezna veza

- Neidentifikuju a ili slaba veza zavisi od na ina definisanja klju a eva od "roditelja" ka "detetu" na dva na ina:
  - kao obavezna neidentifikuju a veza i
  - kao neobavezna (opciona) neidentifikuju a veza.
- Ako je veza (relationship) obavezna (No Nulls ili Mandatory) iz perspektive "roditelj", onda je "dete" egzistencijalno zavisno od "roditelja". No nulls ili Mandatory zna i da je obavezan unos prenesenog klju a entiteta "roditelj" u okviru entiteta "dete" (Klju entiteta A (FK)).
- Ako je veza neobavezna (Nulls Allowed ili Optional), tada "dete" niti je egzistencijalno niti identifikaciono zavisno, ali poštuje tu vezu. Null Allowed ili Optional zna i da nije obavezan (može biti Null) unos prenesenog klju a entiteta "roditelj" u okviru entiteta "dete" (Klju entiteta A (FK)).



## Asocijativni entiteti

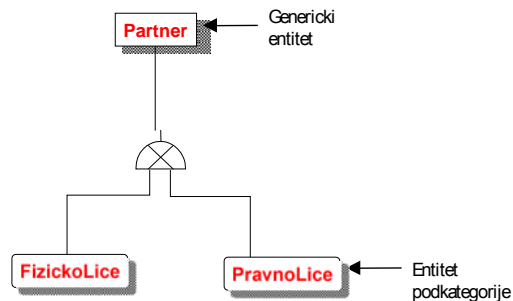
- *Asocijativni entiteti* su sastavljeni od više veza izme u dva ili više entiteta, kao što se može videti na slede oj slici. Npr., ako Magacin ima više PredmetaPoslovanja i jedan PremetPoslovanja se nalazi u više Magacina, tada je Stanje asocijativni entitet koji opisuje vezu izme u entiteta: Magacin i PredmetPoslovanja. Dakle, asocijativni entiteti nose informaciju o višezna noj vezi.



Slika. Veza asocijativnog entiteta Stanje sa nezavisnim entitetima Magacin i PredmetPoslovanja

## Podkategorije entiteta

- Entitet kategorija (category) zavisan je entitet, koji ima tzv. vezu tipa potkategorije (sub-category). Kod entiteta tipa kategorije definišu se: nadređeni entitet, koji ima zajedničke osobine (npr. entitet Partner) i podređeni entiteti (entiteti: FizickoLice i PravnoLice), koji se identifikuju ključem nadređenog i poseduju svoje specifične osobine.

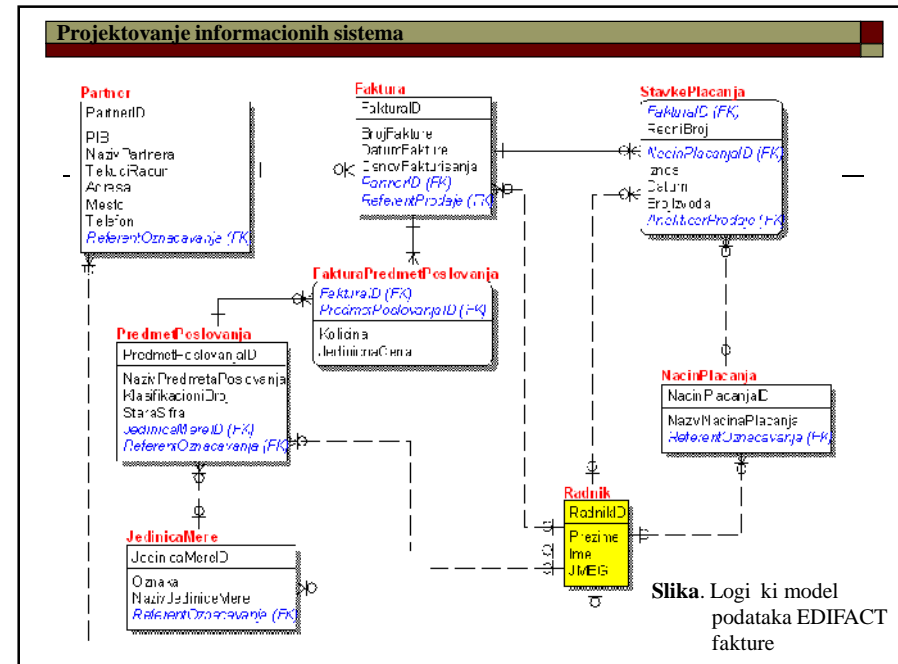


Slika. Primer potkategorije entiteta

- EDIFACT standard, definisan dokumentom UN/ECE WP.4, preporuka broj 6, izdanje iz 1975. godine, preporuče:
  - obrazac za fakturu u međunarodnoj trgovini prema ISO 6422 (skraćeno: UNLK) – **JUS /ISO 6422**,
  - obrazac je baziran na principu "box design",
  - razmeštaj rubrika je fiksiran kao npr. **adresa primaoca**, koja je definisana zbog koverata sa prozorom
  - razmatrani su administrativni, pravni i trgovinski aspekti,
  - prostor "za slobodno raspolaganje" u najnižem delu obrasca namenjen je za više posebnih potreba za individualne aplikacije,
  - ako trgovski partneri primenjuju dokumente koji sadrže šire polje podataka nego set UNLK, ili od onog što je propisano u standardima za odgovarajuće podatke, o tome se moraju posebno dogovoriti.
- Sve ovo je dovelo da se prihvati jedan standardizovani obrazac (**ISO 7372**), **EDIFACT fakture**.

| Projektovanje informacionih sistema                       |                  |  |       |                          |
|---|------------------|--|-------|--------------------------|
| SPONZORILAC<br>STEVAN RADOVIC<br>BEOGRAD<br>11000 BEOGRAD |                  | DAJILJIBNA FAKTURA<br>BROJ: 12345<br>DNEV: 15.05.2024<br>FAKTUR<br>PARTNER |       |                          |
| PRIMALAC<br>ZEMUN<br>11000 BEOGRAD                        |                  | KUPAC (AGENCIJA PRIMALAC)  |       |                          |
| ZEMUN<br>11000 BEOGRAD                                    |                  | ZEMUN, POBROJA, USA<br>USLUGI ISPOSTARSKO PLAGANJA                         |       |                          |
| OTPRIMNE OZNAKE   | BRIGI USTRAJANJE | KOD  | KG    | ZAPREMINA m <sup>3</sup> |
| OPIS ARTIKLA (SIFRA I/II NAZIV)                           | KOLIČINA         | J. CENA  | IZNOS |                          |
| LESENO STANJE   | 3                | 1200   | 3600  |                          |
| TRACIJA   | 2                | 1000   | 2000  |                          |
| PACIJAR   | 1                | 2500   | 2500  |                          |
| AMBALAZA  |                  |  |       |                          |
| PREGVOZ   |                  |  |       |                          |
| OSTALI TRGOVNI  |                  |  |       |                          |
| USLUGOVI  |                  |  |       |                          |
| UKUPNO  |                  | 6000   |       |                          |

Obrazac EDIFACT fakture



## CRUD matrice

- ❑ Kvalitet sinhronizacije podrazumeva da svaki objekat treba da ima najmanje jedno kreiranje (*C – create*), jedno išitavanje (*R – read*), jedno menjanje ili modifikovanje (*U – update*) i jedno brisanje (*D – delete*) da bi sistem bio kompletan.
- ❑ CRUD matrice dokumentuju ove zahteve i sinhronizuju modele podataka, procesa i mreža.

## Sinhronizacija modela procesa i modela podataka

Podatak-Proces-CRUD matrica

| Objekat<br>Atribut                | Proces<br>aplikacije<br>klijenta | Proces<br>promene<br>adrese<br>klijenta | Proces<br>nova<br>porudzbine<br>klijenta | Proces<br>ponistavanje<br>porudzbine<br>klijenta | Proces<br>promene<br>porudzbine<br>klijenta | Proces<br>promene<br>cene<br>proizvoca |
|-----------------------------------|----------------------------------|---|--|--|---|--|
| Klijent                           | C                                |   | R  | R  | R   |  |
| .Šifra klijenta                   | C                                |   | R  | R  | R   |  |
| .Naziv klijenta                   | C                                | U                                       | R  |  | R   |  |
| .Adresa klijenta                  | C                                | U                                       | R  |  | RU  |  |
| .Iznos kredita klijenta           |                                  |   | R  |  | R   |  |
| Porudzbina                        |                                  |   | C  | D  | RU  |  |
| .Šifra porudzbine                 |                                  |   | C  |  | R   |  |
| .Datum porudzbine                 |                                  |   | C  |  | J   |  |
| .Iznos porudzbine                 |                                  |   | C  |  | J   |  |
| Naručen proizvod                  |                                  |   | C  | D  | CRUD  |  |
| .Količina porudzbine              |                                  |   | C  |  | CRUD  |  |
| .Jedinična cena naručenog artikla |                                  |   | C  |  | CRUD  |  |
| Proizvod                          |                                  |   | R  | R  | R   | RU                                     |
| .Šifra proizvoda                  |                                  |   | R  | R  | R   |  |
| .Naziv proizvoda                  |                                  |   | R  |  | R   |  |
| .Opis proizvoda                   |                                  |   | R  |  | R   |  |
| .Jedinična cena proizvoda         |                                  |   | R  |  | R   | RU                                     |
| .Jedinična cena proizvoda         |                                  |   | R  |  | R   | U                                      |

C - kreirati (create)    R - iščitavati (read)    U - promeniti (update)    D - obrisati (delete)

# Analiza modela podataka

- Analiza modela podataka
- Pojam baze podataka
- Sistem za upravljanje bazom podataka
- Normalizacija relacione baze podataka

- **Analiza modela podataka:**
  - ~~tehnika modela podataka u fazi pripreme za fizički dizajn baze podataka za poboljšanje:~~
    - strukture koja umanjuje fleksibilnost i skalabilnost ili
    - strukture koja stvara nepotrebnu redundantnost.
- **Kriterijumi** za kreiranja *dobrog* modela podataka:
  - *jednostavnost* – atributi jednog objekta treba da opisuju samo taj objekat,
  - *neredundantnost* – da model nema ponovljenih atributa,
  - *fleksibilnost i skalabilnost (adaptibilnost)* za buduće potrebe.
- **Normalizacija baze podataka:**
  - proces koji priprema model podataka za implementaciju *jednostavne, neredundantne, adaptibilne* baze podataka.
  - eliminiše redundantnost u atributima i omogućava lakše održavanje integriteta podataka i jednostavniju manipulaciju.

**Baza podataka (BP):**

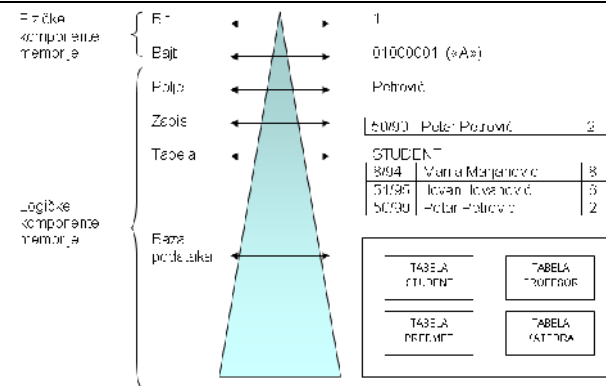
- kolekcija međusobno povezanih podataka, uskladištenih sa minimumom redundanse, koje koriste, zajednički, svi procesi obrade u sistemu.
- samo-opisujuća kolekcija integriranih *zapisa*
- *zapis* je reprezentacija fizičkog, ili konceptualnog objekta
- sa aspekta implementacije predstavlja *skup tabela* međusobno povezanih putem spoljnog ključa.



6.11.2015

Poslovni informacioni sistemi

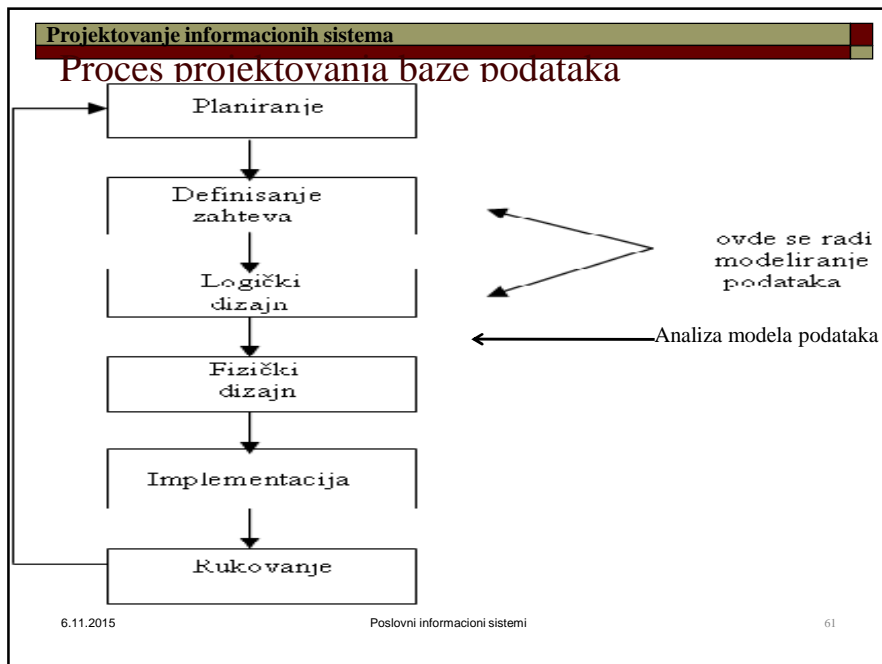
59



6.11.2015

Poslovni informacioni sistemi

60



Projektovanje informacionih sistema

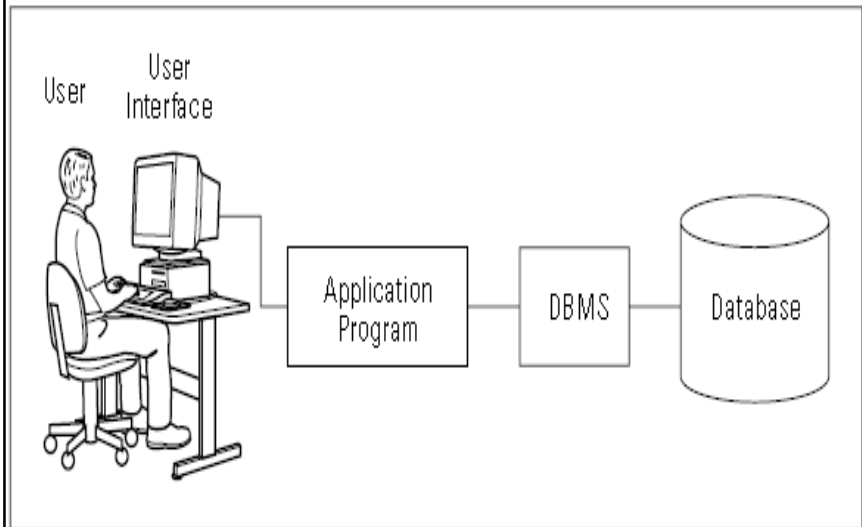
### Sistem za upravljanje bazom podataka- DBMS

□ **DBMS (Data Base Management Systems):**

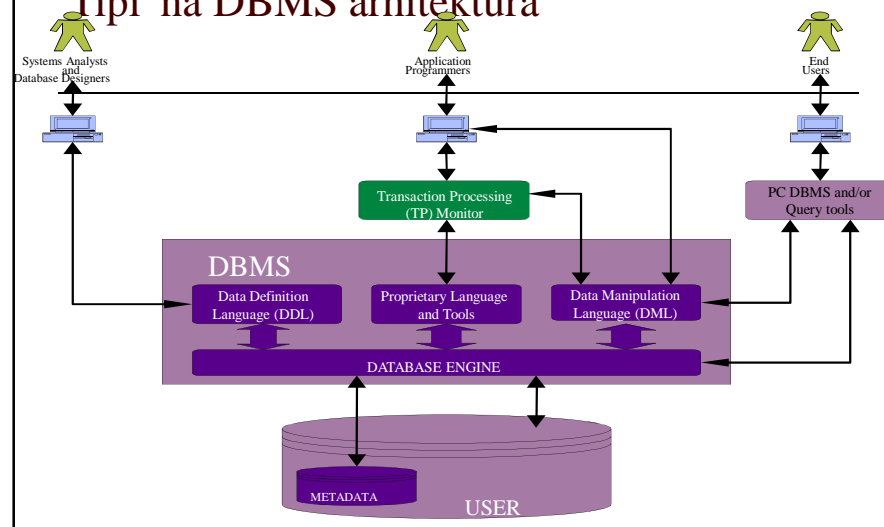
- *Skup programa za definisanje, administraciju i procesiranje baza podataka i pridruženih aplikacija,*
- *Struktura izgrađena za održavanje dragocenih podataka,*
- *Alat koji se koristi za izgradnju strukture i operacije nad podacima sadržanim u bazi podataka – vrednost nije u podacima nego u strukturi podataka!!!!??*
- *Trend je da se razvije skalabilna DBMS koja radi na više različitih platformi.*
- *Složeni softverski sistem koji treba da omogući:*
  - *skladištenje podataka sa minimumom redundanse,*
  - *korišćenje zajedničkih podataka od strane svih ovlašćenih korisnika,*
  - *logičku i fizičku nezavisnost programa od podataka gde svaki korisnik dobija svoju sopstvenu logičku sliku podataka kakva njemu najviše odgovara, bez obzira što se podaci fizički pamte, po pravilu, samo jednom, u jedinstvenoj fizičkoj organizaciji,*
  - *jednostavno komuniciranje sa bazom podataka preko jezika bliskih korisniku – neprofesionalnom i profesionalnom.*

6.11.2015 Poslovni informacioni sistemi 62

### Blok diagram IS sa DBMS

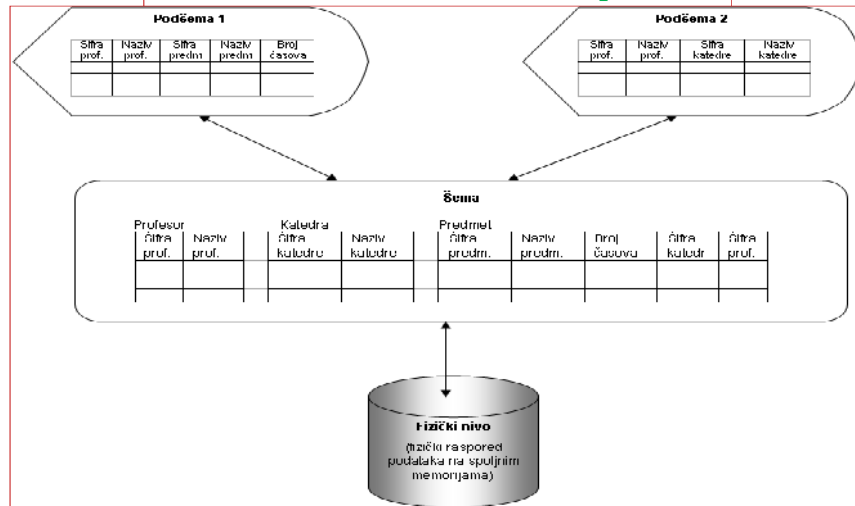


### Tipi na DBMS arhitektura





### Tronivovska struktura DBMS kroz primer



□ *Prema na inu struktuiranja zapisa:*

1. **Relacioni:**

- savremene instalacije DBMS
- implementira podatke u seriji 2-dimenzionalnih tabela
- tabele su u me usobnoj relaciji preko spoljnih klju eva.

2. **Hijerarhijski:**

- imaju jednostavnu hijerarhijsku strukturu
- dopušta brzi pristup podacima
- imaju redundantnost podataka i
- nefleksibilnu strukturu koja otežava modifikaciju baze podataka

3. **Mrežni:**

- imaju minimalnu redundantnost, ali i kompleksnu strukturu.

## Projektovanje informacionih sistema

- Posедуje semanti ki bogatije koncepte za opis strukture i znatno mo nije operacije i attribute od drugih modela.
- Struktura baze podataka može se menjati bez izmene aplikacija baziranih na starim strukturama.

**Primer:** dodavanjem 1 ili više kolona u tabeli DB ne treba menjati ranije razvijene aplikacije, sve dok se ne izmeni 1 ili više kolona koje te aplikacije koriste

- Tabela se može definisati kao matemati ka relacija i zatim iskoristiti bogata teorijska osnova odgovaraju eg matemati kog aparata.
- **Svaka relacija ima svojstva skupa** - osnovno svojstvo svakog skupa:
  - *elementi koje sadrži me usobno se razlikuju,*
  - *tj. svi redovi tabele me usobno se razlikuju.*
- **Svaka relacija je skup**, a svaka tabela nije.
- **Da bi tabela bila relacija mora da zadovolji slede e uslove:**
  - *ne postoje duplikati vrsta tabele*
  - *redosled vrsta i redosled kolona nije zna ajan*
  - *nisu dozvoljni atributi ili grupe atributa sa ponavljanjem, odnosno*
  - *nisu dozvoljene tabele u tabeli.*

6.11.2015

Poslovni informacioni sistemi

67

## Projektovanje informacionih sistema

### □ Šema:

- struktura cele DB ili konceptualni pogled
- esto se naziva kompletni logi ki pogled na DB
- metapodatak, deo DB, uskladišten u tabeli kao svaki drugi podatak

### □ Domen atributa:

- skup svih vrednosti koje može primiti *atribut relacije* tj. *kolona tabele*
- karakteristike kolone tabele plus ograni enja atributa odre uju domen kolone

**Primer:** U tabeli DB "INVENTAR", prodavac automobila ima kolonu BOJA, ako auto ima 4 boje na raspolaganju- *metalik, crvena, bela i žuta*, ove 4 boje su domen atributa BOJA

### □ Ograni enja atributa:

- *strukturna* - odnose se na kardinalnosti preslikavanja
- *vrednosna* - odre uju koje vrednosti atributi tabele mogu imati i spre ava unos nekorektnih podataka u kolone.

6.11.2015

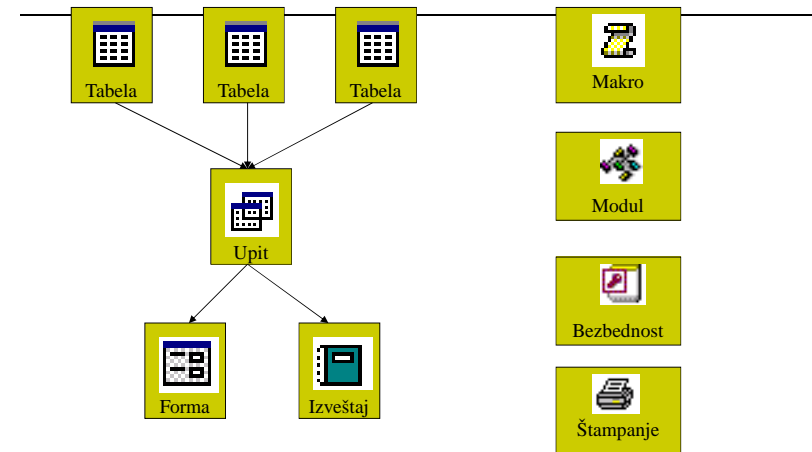
Poslovni informacioni sistemi

68

## MS Access kao DBMS

- Da bi se okvalifikovala kao potpun sistem za upravljanje relacionom bazom podataka, aplikacija mora da izvršava sledeće četiri osnovne funkcije, od kojih svaka ima sopstvenu prezentaciju za korisnika:
- ◆ **Organizacija podataka** – obuhvata izradu i rukovanje tabelama koje sadrže podatke u konvencionalnom tabelarnom formatu koju Access naziva pogled (*Datasheet*).
  - ◆ **Povezivanje tabela i izdvajanje podataka** – povezuje više tabela prema relacijama izme u podataka radi izrade privremenih tabela, koje sadrže izabrane podatke. Access koristi upite da bi povezao tabele i izabrao podatke koji će se učitati u privremenoj tabeli, koja se naziva objekat **Recordset**. Objekti **Recordset** nazivaju se virtuelne tabele, jer se učitavaju u memoriji računara umesto u datotekama baze podataka.
  - ◆ **Unos i uređivanje podataka** – zahteva projektovanje i implementaciju obrazaca za pregled, unos i uređivanje podataka kao alternativu tabelarnom prikazu. Obrasci su ti koji umesto aplikacije omogućavaju da kontrolišete prikazivanje podataka.
  - ◆ **Prikazivanje podataka** – zahteva izradu izveštaja koji mogu da sumiraju podatke u skupovima zapisa (Recordset). Njih možete da pregledate, štampate ili objavljujete na internetu ili intranetu.

## Osnovne funkcije za podršku Accessa



□ **RELACIJA JEDAN-PREMA-JEDAN**

Jednom redu u jednoj tabeli odgovara jedan red u drugoj tabeli. Ovakve tabele možete kombinovati u jednu tabelu koja se sastoji od svih kolona obe tabele.

□ **RELACIJA TIP A JEDAN-PREMA-VIŠE**

Povezuju jedan red iz jedne tabele sa više redova druge tabele preko relacije izme u primarnog ključa i bazne tabele i odgovaraju eg spoljnog ključa u povezanoj tabeli.

□ **RELACIJE TIP A VIŠE-PREMA-JEDAN**

Povezuju više redova jedne tabele sa jednim redom druge tabele.

□ **RELACIJE TIP A VIŠE-PREMA-VIŠE I ETVRTA NORMALNA FORMA**

Ne mogu da se izraze kao jednostavne relacije izme u dva sudeluju a entiteta. Njih ostvarujete tako što pravite tabelu koja ima relacije tipa više-prema-jedan sa dve bazne tabele.

## Normalizacija baze podataka

- **Normalizacija** je formalizovani postupak za grupisanje atributa podataka u tabele i tabele u baze podataka.

□ Ciljevi normalizacije:

- Eliminisanje dupliranih informacija u tabelama.
- Prilagoavanje budućim izmenama u strukturi tabele.
- Umanjivanje uticaja strukturnih izmena baze podataka na korisnike aplikacije koje pristupaju podacima.

□ Pravila normalizacije

- Prva normalna forma zahteva da tabele budu ravne i da ne sadrže duplirane grupe.
- Druga normalna forma zahteva da podaci u svim kolonama koje nisu deo ključa budu potpuno zavisni od primarnog ključa i svakog elementa (kolone) primarnog ključa kada je on složeni primarni ključ. Potpuno zavisni znači da je vrednost podatka u svakoj koloni koja nije deo ključa zapisa, na jedinstven način određena vrednošću u primarnom ključu. Druga normalna forma uklanja veći deo nepotrebnih (redundantnih) podataka.
- Treća normalna forma zahteva da sve kolone koje nisu deo ključa tabele budu zavisne od primarnog ključa tabele i nezavisne jedna od druge. Tabele moraju da odgovaraju prvoj i drugoj formi da bi bile sposobne za treću normalnu formu.

## Projektovanje informacionih sistema

- Access omogućava pravljenje četiri osnovna tipa upita, za postizanje različitih ciljeva:
  - **Upiti za izbor** (*Select Querys*) izdvajaju podatke iz jedne ili više tabela i prikazuju ih u tabelarnom obliku.
  - **Upiti unakrsnih tabela** (*Crosstab queries*) sumiraju podatke iz jedne ili više tabela u obliku radne tabele. Ovakvi upiti su korisni za analiziranje podataka i izradu grafika ili dijagrama, na osnovu sume vrednosti numeričkih polja većeg broja zapisa.
  - **Akcioni upiti** (*Action queries*) prave nove tabele iz tabela upita, ili prave velike izmene u nekoj tabeli. Takvi upiti dopuštaju da dodate ili obrišete zapise iz tabele, ili napravite izmene u zapisima na osnovu izraza koji unosite pri dizajnu upita.
  - **Parametarski upiti** (*Parameter queries*) čije se korišćenje ponavlja pri čemu se vrše samo jednostavne izmene njihovih kriterijuma. Kad izvršavate parametarski upit, Access prikazuje okvir za dijalog koji od vas zahteva da unesete novi kriterijum. Parametarski upiti zapravo nisu poseban tip upita, jer ove parametarske funkcije možete da dodate u upite za izbor, upite unakrsnih tabela i u akcione upite.

## Projektovanje informacionih sistema grupama i korisnicima

- Većina klijent-server baza podataka prepoznaje sledeće tri grupe korisnika baze podataka:
  1. **Administratori** (*Admins*) imaju ovlašćenja da pregledaju i ažuriraju postojeće tabele i dodaju ili obrišu tabele i druge objekte baze podataka iz baze podataka. Članovi grupe Admins obično ne imaju dozvolu da menjaju aplikacije sadržane u bazama podataka.
  2. **Obični članovi radnih grupa** (*Users*) imaju dozvolu da otvore bazu podataka, a po potrebi im se dodeljuje dozvola za pregledanje i menjanje baza podataka.
  3. **Povremenim korisnicima baze podataka** (*Guests*) esto su dodeljena ograničena prava da koriste bazu podataka i objekte koje ona sadrži, ali im se ne dodeljuje korisnički nalog.
- Access ima dva nivoa bezbednosti:
  - **na nivou aplikacije** (zahteva da svaki korisnik Accessa unese korisničko ime i lozinku da bi mogao da pokrene Access) i
  - **na nivou datoteke** (uspostavio je mrežni operativni sistem, kao što je *Windows NT Server* i ona određuje korisnicima dozvoljava ili ne dozvoljava pristup zajedničkim omotnicama i/ili pojedinačnim datotekama).