



**BAZE PODATAKA**

# **Relacioni Model**

## **(Relational Model)**



# Relacioni Model

- Definicija Relacionog modela
- Prevođenje modela E-V u Relacioni model



# Definicija Relacionog modela

- **Relacija**  $r$  je podskup dekartovog proizvoda

$D_1 \times D_2 \times \dots \times D_n$ , za dati skup  $D_1, D_2, \dots, D_n$

Prema tome, relacija je skup  $n$ -torki  $(a_1, a_2, \dots, a_n)$  takvih da  $a_i \in D_i$

- Primer: Ako su dati skupovi

$customer\_name = \{Jones, Smith, Curry, Lindsay\}$

$customer\_street = \{Main, North, Park\}$

$customer\_city = \{Harrison, Rye, Pittsfield\}$

tada je  $r = \{ (Jones, Main, Harrison), (Smith, North, Rye), (Curry, North, Rye), (Lindsay, Park, Pittsfield) \}$

relacija na

$customer\_name \times customer\_street \times customer\_city$



# Atributi

- Svakom **atributu** u relaciji se dodeljuje ime
- Skup dozvoljenih vrednosti svakog atributa se naziva **domen** atributa
- Vrednosti atributa su (uobičajeno) **atomske** ili nedeljive
  - Primedba: vrednosti viševrednosnih atributa nisu atomske
  - Primedba: vrednosti složenih atributa nisu atomske
- ***null*** vrednosti pripadaju svakom domenu
- ***null*** vrednosti nisu poželjne



# Relaciona Šema

- $A_1, A_2, \dots, A_n$  su *atributi*
- $R = (A_1, A_2, \dots, A_n)$  je **Relaciona Šema**  
specificira ime relacije, ime i tip svake kolone (atributa).
  - Primer: **Students**(sid: string, name: string, login: string, age: integer, gpa: real)
  - Primer: **Customer\_schema** = (*customer\_name*,  
*customer\_street*,  
*customer\_city*)
- $r(R)$  je *relacija sa šemom R*  
Primer:  
*customer* (**Customer\_schema**)



# Relaciona Instanca

- **Relaciona Instanca** predstavlja tekuće vrednosti relacije koje se predstavljaju tabelom
- Element ***t*** relacije ***r*** se naziva **torka**, i predstavljen je vrstom u tabeli
- Sve torke (vrste) su različite

<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>
<i>Jones</i>	Main	Harrison
<i>Smith</i>	North	Rye
<i>Curry</i>	North	Rye
<i>Lindsay</i>	Park	Pittsfield

atributi  
(ili kolone)

torke  
(ili vrste)

relacija *customer*



## Relacije su neuređene

- Redosled torki je nebitan (torke se mogu smeštati proizvoljnim redosledom)
- Primer: relacija *account* sa neuređenim torkama

<i>account_number</i>	<i>branch_name</i>	<i>balance</i>
A-101	Downtown	500
A-215	Mianus	700
A-102	Perryridge	400
A-305	Round Hill	350
A-201	Brighton	900
A-222	Redwood	700
A-217	Brighton	750



# Relaciona Baza

- Relacionu Bazu čini skup relacija
- Informacije o okruženju koje modeliramo su razbijene na relacije, tako da svaka relacija sadrži informacije o nekom delu okruženja, npr.
  - account* : sadrži informacije o računima
  - depositor* : sadrži informacije o tome koji klijent poseduje koji račun
  - customer* : sadrži informacije o klijentima
- Sve informacije se mogu nalaziti u jednoj relaciji, npr.  
*bank* (*account\_number*, *balance*, *customer\_name*, ..)  
Takav pristup dovodi do
  - ponavljanja informacija (npr., dva klijenta poseduju račun)
  - potreba za null vrednostima (npr., ako klijent nema račun)





# Relacija Klijent (Customer)

<i>customer_name</i>	<i>customer_street</i>	<i>customer_city</i>
Adams	Spring	Pittsfield
Brooks	Senator	Brooklyn
Curry	North	Rye
Glenn	Sand Hill	Woodside
Green	Walnut	Stamford
Hayes	Main	Harrison
Johnson	Alma	Palo Alto
Jones	Main	Harrison
Lindsay	Park	Pittsfield
Smith	North	Rye
Turner	Putnam	Stamford
Williams	Nassau	Princeton



# Relacija Ulagač (Depositor)

<i>customer_name</i>	<i>account_number</i>
Hayes	A-102
Johnson	A-101
Johnson	A-201
Jones	A-217
Lindsay	A-222
Smith	A-215
Turner	A-305



# Ključ

- Neka je  $K \subseteq R$
- $K$  je **superključ**  $R$  ako dve torke ne mogu imati iste vrednosti za  $K$  u relaciji  $r(R)$ 
  - Primer:  $\{customer\_name, customer\_street\}$  i  $\{customer\_name\}$   
su superključevi za *Customer*, ako dva klijenta ne mogu imati ista imena.
- $K$  je **kandidat ključ** ako je superključ, a nijedan podskup  $K$  nije superključ  
Primer:  $\{customer\_name\}$  je kandidat ključ za *Customer*, jer je superključ (ako dva klijenta ne mogu imati ista imena), i nijedan podskup atributa nije superključ.
- **Primarni Ključ** je izabrani kandidat ključ



# Koraci u dizajnu baze

- **Specifikacija Zahteva**
  - potrebe korisnika; koje zahteve baza mora da zadovolji?
- **Konceptualni Dizajn**
  - konceptualni opis (obično model E-V)
- **Logički Dizajn**
  - prevođenje modela E-V u model konkretnog DBMS
- **Poboljšanja Šema**
  - konzistentnost, normalizacija
- **Fizički Dizajn**
  - indeksi, disk layout
- **Security Dizajn**
  - ko, čemu i kako može pristupiti



# Konceptualni dizajn

- Šta su *entiteti* i *veze* u okruženju?
- Koje informacije o uočenim entitetima i vezama treba čuvati u bazi podataka?
- Koja *ograničenja integriteta* ili *poslovna pravila* treba da budu zadovoljena?
- Grafička predstava (dijagram E-V) šeme baze u modelu E-V.



# Pretvaranje modela E-V u Relacioni

- Slična struktura. Skupovi entiteta i veza se mogu predstaviti kao relacione šeme.
- Svaki skup entiteta i veza se može predstaviti jedinstvenom relacionom šemom sa imenom koje odgovara imenu relevantnog skupa entiteta ili veza.
- Svaka relaciona šema ima kolone jedinstvenih imena koje odgovaraju atributima.
- Baza podataka koja odgovara dijagramu E-V može se predstaviti skupom relacionih šema.



## Skupovi Entiteta u Relacije (Tabele)

- **Skup jakih entiteta** se prevodi u relacionu šemu (tabelu) sa istim atributima u kolonama.
- **Skup slabih entiteta** postaje tabela koja, pored kolona za attribute skupa slabih entiteta ima i kolone za svaki atribut primarnog ključa identifikacionog skupa jakih entiteta

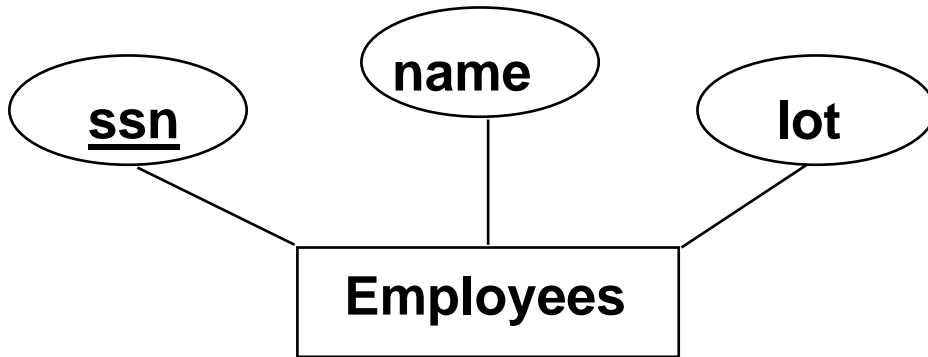
Primer:

*payment* =

*(loan\_number, payment\_number, payment\_date, payment\_amount)*



# Skupovi Entiteta u Relacije (Primer)



ssn	name	lot
123-22-3666	Attishoo	48
231-31-5368	Smiley	22
131-24-3650	Smethurst	35





## Skupovi Veza u Tabele (1)

- **Skup Veza** sa ograničenjem više-u-više se predstavlja šemom sa:
  - 1) atributima koji odgovaraju atributima primarnih ključeva skupova entiteta koji su u vezi, i
  - 2) atributima koji odgovaraju opisnim atributima skupa veza.

Primer: Šema za skup veza *borrower*

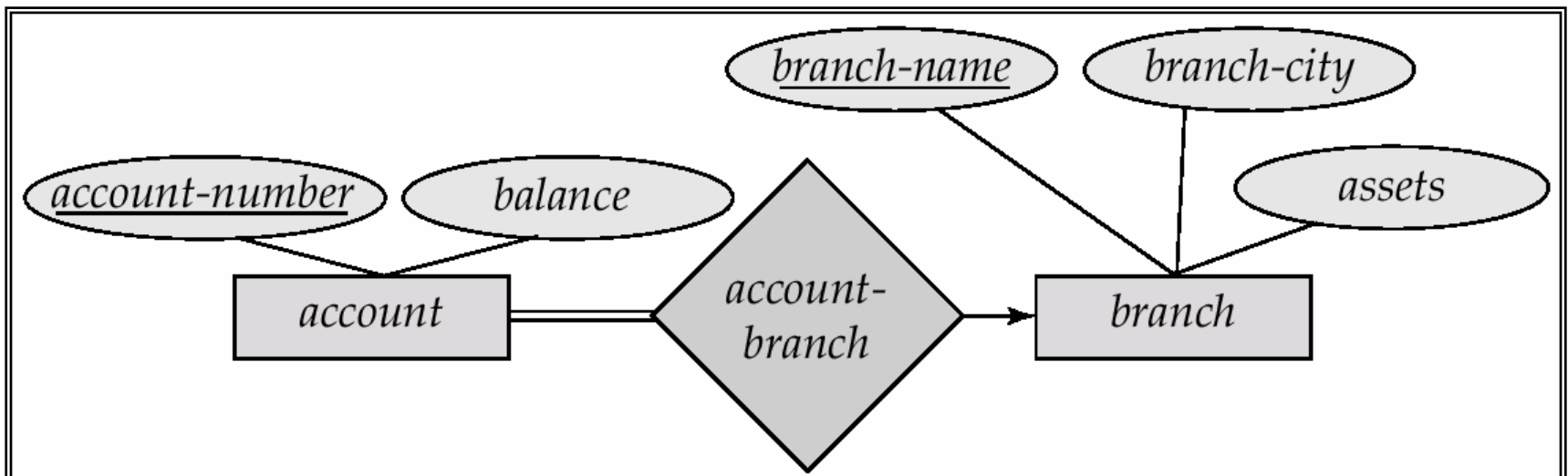
*borrower* = ( *customer id*, *loan number* )



## Skupovi Veza u Tabele (2)

- Skupovi veza sa ograničenjem više-u-jedan i jedan-u-više koji su totalni na strani "više" mogu se predstaviti dodavanjem ekstra atributa relacijama nastalim od skupova entiteta na strani "više", koji odgovaraju atributima primarnog ključa na strani "jedan"

Primer: Umesto kreiranja šeme za skup veza *account\_branch*, dodaje se atribut *branch\_name* u šemu nastalu od skupa entiteta *account*





## Skupovi Veza u Tabele (3)

- Za skupove veza jedan-u-jedan, bilo koja strana može biti odabrana da bude strana "više"
  - Prema tome, ekstra atribut može biti dodat bilo kojoj tabeli koja odgovara skupovima entiteta koji su u vezi
- Ako je participacija *parcijalna* na strani "više", dodavanje ekstra atributa u šemu na strani "više" može dovesti do pojave null vrednosti
- Šema koja odgovara skupu veza koji povezuje skupove slabih i njima identifikacionih skupova jakih entiteta je redundantna.
  - Primer: Šema *payment* već sadrži attribute koji bi se pojavili u šemi *loan\_payment* (tj., attribute *loan\_number* i *payment\_number*).



# Složeni i atributi sa više vrednosti

- **Kompozitni atributi se predstavljaju kreiranjem posebnih atributa za svaku komponentu**
  - Primer: Šema koja odgovara skupu entiteta *customer* sa kompozitnim atributom *name* sa komponentama *first\_name* i *last\_name* ima dva atributa  
*name.first\_name* i *name.last\_name*
- **Viševrednosni atribut  $M$  skupa entiteta  $E$  se predstavlja posebnom šemom  $EM$** 
  - Šema  $EM$  ima attribute koji odgovaraju primarnom ključu  $E$  i atribut koji odgovara viševrednosnom atributu  $M$   
Primer: Viševrednosni atribut *dependent\_names* skupa entiteta *employee* se predstavlja šemom:  
*employee\_dependent\_names* = ( *employee\_id*, *dname* )
  - **Svaka vrednost viševrednosnog atributa se mapira u posebnu toroku relacije sa šemom  $EM$**   
Na primer, entitet *employee* sa primarnim ključem 123-45-6789 i *dependents* Jack i Jane se mapiraju u dve torke:  
(123-45-6789 , Jack) i (123-45-6789 , Jane)



# Predstavljanje Specijalizacije (1)

- Metod 1:
  - Formirati šemu za skupove entiteta višeg nivoa
  - Formirati šemu za svaki skup entiteta nižeg nivoa, koja uključuje primarni ključ skupa entiteta višeg nivoa i lokalne attribute

šema

*person*

*customer*

*employee*

atributi

*name, street, city*

*name, credit\_rating*

*name, salary*

Nedostatak: ako želimo da dobijemo informaciju o *employee* potrebno je pristupiti dvema relacijama, jednoj koja odgovara šemi nižeg nivoa i jednoj koja odgovara šemi višeg nivoa



## Predstavljanje Specijalizacije (2)

- Metod 2:
  - Formirati šemu za svaki skup entiteta sa svim lokalnim i nasleđenim atributima

*schema*                      *attributes*

*person*                      *name, street, city*

*customer*                      *name, street, city, credit\_rating*

*employee*                      *name, street, city, salary*

- Ako je specijalizacija totalna, šema za generalizovani skup entiteta (*person*) se ne koristi za čuvanje informacija
  - Može biti definisana kao "view" koji sadrži uniju specijalizovanih relacija
  - Ali eksplicitna šema može biti potrebna zbog stranog ključa

Nedostatak: *street* i *city* mogu biti redundantni za osobe koje su i klijenti i zaposleni



# Predstavljanje Agregacije (1)

- Za predstavljanje agregacije, treba kreirati šemu koja sadrži
  - primarni ključ agregirane veze,
  - primarni ključ pridruženog skupa entiteta
  - opisne attribute



## Predstavljanje Agregacije (2)

- Na primer, za predstavljanje agregacije *manages* između veza *works\_on* i skupa entiteta *manager*, treba kreirati šemu *manages* (*employee\_id*, *branch\_name*, *title*, *manager\_name*)
- Šema *works\_on* je redundantna i mogu se pojaviti null vrednosti za atribut *manager\_name* u relaciji sa šemom *manages*

