

# **RELACIONI MODEL**

# RELACIONI MODEL

Dve karakteristike čine relacioni model još uvek najpopularnijim i najšire primenjivanim:

- Struktura modela je veoma jednostavna, prihvatljiva svakom korisniku, jer relacionala baza podataka predstavlja skup tabela. I same operacije, koje iz skupa datih tabela generišu novu, su jednostavne i lako prihvatljive.
- Moguća je formalno-matematička interpretacija tabela. Određene vrste tabela se mogu tretirati kao matematičke relacije i zatim iskoristiti bogata teorijska osnova odgovarajućeg matematičkog aparata.

# STRUKTURA RELACIONOG MODELA

- **Skup.** Pojam skupa je osnovni pojam u matematici i on se formalno ne definiše. Neformalno se može reći da je skup "objedinjavanje nekog mnoštva elemenata u celinu".

Skup elemenata  $x, y, z, \dots, v$  se može označiti sa  $S = \{x, y, z, \dots, v\}$ .

- **Ekstenzija skupa** - Navodjenje svih elemenata skupa
- **Intenzija skupa** - Navodjenje osobina koje svaki element skupa mora da zadovolji:  
 $S = \{x \mid P(x)\},$   
gde je  $P(x)$  uslov koji svi elementi treba da zadovolje

# STRUKTURA RELACIONOG MODELA

- Dekartov (Kartezijanski) proizvod skupova. Neka je data kolekcija skupova  $D_1, D_2, \dots, D_n$  (ne neophodno različitih). Dekartov proizvod ovih  $n$  skupova

$$D_1 \times D_2 \times \dots \times D_n$$

je skup svih mogućih uređenih  $n$ -torki

$$(d_1, d_2, \dots, d_n),$$

tako da je  $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$ .

*Primer.*  $A = \{1, 2, 3, 4\}, B = \{4, 6, 8\}$

$$A \times B = \{(1,4), (1,6), (1,8), (2,4), (2,6), (2,8), (3,4), (3,6), (3,8), (4,4), (4,6), (4,8)\}.$$

# STRUKTURA RELACIONOG MODELA

- Relacija koja najopštije predstavlja odnos između elemenata nekih ne neophodno različitih skupova definiše se kao podskup Dekartovog proizvoda tih skupova.

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

(Podskup sadrži one n-torke Dekartovog proizvoda koje zadovoljavaju odnos koji relacija predstavlja)

*Primer:* Neka je na skupovima A i B iz Primera 1 zadata relacija

$$R \subseteq A \times B = \{(a,b) \mid a=b/2\}$$

Na osnovu definicije relacije očigledno je da je

$$R = \{(2,4), (3,6), (4,8)\}$$

# STRUKTURA RELACIONOG MODELA

- **Domen relacije.** Skupovi  $D_1, D_2, \dots, D_n$  se nazivaju domenima relacije  $R$ .
- Broj domena na kojima je definisana neka relacija se naziva **stepen relacije**.
- Kardinalnost relacije je broj  $n$ -torki u relaciji.

# STRUKTURA RELACIONOG MODELA

- Atribut relacije se formalno može definisati kao preslikavanje:

Atribut: n\_torka\_relacije ---> Domen

odnosno kao par (**naziv domena, vrednost domena**)

Ovakva definicija atributa omogućava da se **relacija predstavi kao tabela.**

# STRUKTURA RELACIONOG MODELA

$\text{STUDENT} \subseteq \text{BrInd} \times \text{Ime} \times \text{Starost} =$   
 $\{ \langle 001, \text{Miloš}, 19 \rangle, \langle 007, \text{Ana}, 19 \rangle, \langle 035, \text{Ana}, 22 \rangle \}$

(REDOSLED N-TORKI JE BITAN)

BRIND	IME	STAROST
001	M ilo š	19
007	A n a	19
035	A n a	35



# STRUKTURA RELACIONOG MODELA:

## PRVA NORMALNA FORMA

- Definišu se sledeći uslovi koje tabela mora da zadovolji da bi bila relacija:
  - (1) Ne postoje duplikati vrsta tabele;**
  - (2) Redosled vrsta nije značajan;**
  - (3) Redosled kolona nije značajan.**
  - (4) Sve vrednosti atributa u relacijama su atomske,** ili drugim rečima, nisu dozvoljeni atributi ili grupe atributa "sa ponavljanjem"), odnosno nije dozvoljeno da vrednosti nekih atributa u relaciji budu relacije (nisu dozvoljene "tabele u tabeli").
- Ako relacija zadovoljava navedene uslove tada je ona u **Prvoj normalnoj formi (1NF).**

# NENORMALIZOVANA I NORMALIZOVANA RELACIJA

BRIND	IME	NAZIVPRED	OCENA
001	Miloš	Baze podataka	9
		Matematika	7
		Fizika	10
007	Ana	Fizika	7
		Matematika	9

BRIND	IME	NAZIVPRED	OCENA
001	Miloš	Baze podataka	9
001	Miloš	Matematika	7
001	Miloš	Fizika	10
007	Ana	Fizika	7
007	Ana	Matematika	9

# ŠEMA RELACIJA

Uobičajeno je se relacije predstavljaju na sledeći način koji se naziva i "šema relacija":

```
STUDENT(BRIND, IME, STAROST)  
STUD_ISPIT(BRIND,IME,NAZPRED,OCENA)
```

Naziv relacije se navodi ispred zagrade, a nazivi atributa se navode u zagradi.

**Relaciona baza podataka je kolekcija vremenski promenljivih relacija.**

# DOMENI RELACIJE

Domeni su skupovi iz kojih atributi relacije uzimaju svoje vrednosti. Uobičajeno je da se domeni podele na:

- **Predefinisane domene** (tipove podataka), koji postoje u jezicima baze podataka (**int, char**).
- **Semantičke domene**, koji se još nazivaju apstraktnim ili korisnički definisanim domenima. (Apstraktni tipovi podataka, odnosno objekti koje definiše korisnik).

Atributi relacija bi trebalo uvek da budu definisani nad semantičkim domenima, jer samo tako korisnik može imati punu kontrolu u izvršavanju operacija nad bazom podataka.

# DOMENI RELACIJE

- Dva atributa u modelu su sematički ekvivalentna samo ako su definisana nad istim domenom. Sintaksna konstrukcija za definisanje semantičkog domena mogla bi da ima konstrukciju:

```
CREATE DOMAIN naziv domena  
[(ograničenje)] [(lista operacija)];
```

- Većina komercijalnih relacionih SUBP ne podržava koncept semantičkog domena, ***Objektno-relacione baze podataka*** ga podržavaju i one su obrađene u posebnom poglavlju.

# KLJUČEVI RELACIJE

- Činjenica da su sve n-torke u relaciji različite, govori da postoji jedan atribut (**prost ključ**) ili više atributa zajedno (**složen ključ**) čije vrednosti jedinstveno identifikuju jednu n-torku u relaciji (jednu vrstu u tabeli).

Ključ relacije R je takva kolekcija K njenih atributa koja zadovoljava sledeća dva uslova:

- Osobina jedinstvenosti.** Ne postoje bilo koje dve n-torke sa istom vrednošću K.
- Osobina neredundantnosti.** Ako se bilo koji atribut izostavi iz K, gubi se osobina jedinstvenosti

# KLJUČEVI RELACIJE

- Može u jednoj relaciji postojati više različitih kolekcija  $K$  atributa koje zadovoljavaju definiciju ključa. Sve takve kolekcije se nazivaju kandidati za ključ. Jedan od kandidata koji se izabere da praktično služi za identifikaciju  $n$ -torke relacije se tada naziva primarni ključ. Ostali (neizabrani) kandidati se tada nazivaju alternativnim ključevima.
- Atributi koji učestvuju u ključevima (koji su deo kandidata za ključ) nazivaju se ključnim atributima. Ostali atributi u relaciji su neključni (ili sporedni) atributi.

# SPOLJNI KLJUČ

- Spoljni ključ je atribut (ili grupa atributa) u relaciji R1 koji u njoj nije primarni ključ, ali je primarni ključ u nekoj drugoj relaciji baze podataka. Vrednost spoljnog ključa relacije R1, koristi se za povezivanje sa vrednošću primarnog ključa u odgovarajućoj relaciji R2.

Neka je R2 bazna relacija. Spoljni ključ SK u R2 je poskup njenih atributa takav da:

- postoji bazna relacija R1 sa kandidatom za ključ KK
- svaka vrednost SK u relaciji R2 jednaka je nekoj vrednosti KK u relaciji R1



# RELACIONA BAZA PODATAKA

```
STUDENT(BRIND,MLB,IME,STAROST,POL,ŠIFSˇSMER)  
PREDMET(ŠIFPRED, NAZIVPRED, BROJČASOVA)  
PRIJAVA(BRIND, ŠIFPRED, DATUMPOL, OCENA)  
SMER(ŠIFSˇSMER, NAZIVSMERA, ŠN)  
NASTAVNIK(ŠN, IMENAST,ZVANJE,ŠIFPRED)
```

Relacije u nekoj bazi podataka mogu se podeliti na "bazne" i "izvedene". Izvedena relacija (pogled) je relacija koja se može izvesti iz skupa datih baznih i izvedenih relacija, preko operacija koje se definišu nad relacijama.

# NULA VREDNOSTI

- Termin "nula vrednost" (koji ćemo obeležavati sa ?) se koristi da označi "nedostatak informacija u bazi, odnosno "još nepoznatu vrednost" za neki atribut u nekim n-torkama relacija u relacionoj bazi podataka.

B R I N D	I M E	S T A R O S T
001	M i l o š	19
007	A n a	?
035	?	35

# NULA VREDNOSTI

- Pored nula vrednosti koja ima smisao "još nepoznata vrednost", može se uvesti i druga nula vrednost koja ima smisao "neprimenljivo svojstvo". Ona treba da omogući da se iskaže činjenica da je neki atribut neprimenljivo svojstvo za neka pojavljivanja objekata predstavljenih n-torkama date relacije.
- Relacioni SUBP ili treba da podrže obe "nula vrednosti", ili projektovanjem relacija treba izbeći "nula vrednosti" sa smislom "neprimenljivo svojstvo"

# DINAMIČKA PRAVILA INTEGRITETA

- Pravila integriteta definišu dozvoljena stanja i dozvoljene prelaze sistema iz stanja u stanje. Pravilo integriteta u relacionom modelu se iskazuje definisanjem ograničenja na vrednosti atributa i akcijama koje se preduzimaju kada neka operacija ažuriranja naruši posmatrano ograničenje. Opšta sintaksna konstrukcija za iskazivanje pravila integriteta je:

```
CREATE INTEGRITY RULE naziv pravila  
    ograničenje | naziv_ograničenja  
    [ (ON ATTEMPTED VIOLATION akcija)  
    | (zapeta_lista parova operacija ažuriranja, akcija)]
```

# OGRANIČENJA I PRAVILA INTEGRITETA

- Uobičajeno je da se u relacionom modelu definišu dve vrste pravila integriteta:
  - **Pravila integriteta modela**, koja su posledica strukture relacinog modela, pa su zbog toga opšta i moraju da važe u svakom konkretnom relacionom modelu.
  - **Poslovna pravila integriteta**, odnosno specifična ograničenja za dati relacioni model. Naziv "poslovna" proističe iz činjenice da se preko ovih ograničenja iskazuju specifični odnosi vrednosti atributa koji važe u datom realnom (najčešće poslovnom) sistemu.

# PRAVILA INTEGRITETA MODELA

- Definišu se dva opšta pravila integriteta relacionog modela:

**(1) Integritet entiteta (integritet ključa):** *Ni jedan atribut koji je primarni ključ ili deo primarnog ključa neke bazne relacije ne može da uzme nula vrednost.*

**(2) Referencijalni integritet.** *Ako neka bazna relacija (recimo R2) poseduje spoljni ključ (recimo SK) koji ovu relaciju povezuje sa nekom drugom baznom relacijom (recimo R1), preko primarnog ključa (recimo PK), tada svaka vrednost SK mora biti bilo jednaka nekoj vrednosti PK, ili biti nula vrednost. Relacije R1 i R2 ne moraju biti različite.*

# REFERENCIJALNI INTEGRITET

## FORMALNA DEFINICIJA REFERENCIJALNOG INTEGRITETA:

FOREIGN KEY ( lista atributa koji ga čine)  
REFERENCES naziv rel. primarnog ključa  
DELETE opcija  
UPDATE opcija  
opcija :: RESTRICTED | CASCADES  
| NULLIFIES | DEFAULT

# REFERENCIJALNI INTEGRITET

- **RESTRICTED.** Operacija se odbija ako narušava referencijalno ograničenje.
- **CASCADES.** "Okida" se odgovarajuća operacija i na relaciji u kojoj se nalazi spoljni ključ, da bi se, na taj način, zadovoljilo ograničenje.
- **NULLIFIES.** Zamenjuje se vrednost spoljnog ključa sa "nula vrednošću" za sve n-torke kojima odgovara izbačeni ili promenjeni spoljni ključ.
- **DEFAULT.** Zamenjuje se vrednost spoljnog ključa sa predefinisanom "default vrednošću" za sve n-torke kojima odgovara izbačeni ili promenjeni spoljni ključ.



# POSLOVNA PRAVILA INTEGRITETA

Uobičajeno je da se ova pravila integriteta podele na sledeće podtipove:

- ***Pravila integriteta za domene***, preko kojih se specifikuje koje vrednosti postoje u domenu;
- ***Pravila integriteta za attribute***, preko kojih se definišu dozvoljene vrednosti nekog atributa nezavisno od vrednosti drugih atributa u bazi;
- ***Pravila integriteta za relacije***, preko kojih je moguće vezati vrednost jednog, za vrednost drugog atributa u jednoj relaciji;
- ***Pravila integriteta za bazu***, preko kojih je moguće povezati vrednosti atributa iz više relacija.

# PRAVILA INTEGRITETA ZA DOMENE

Definisanje pravila integriteta za domene svodi se na definisanje semantičkog domena. Pri tome se može koristiti sledeća sintaksa:

```
CREATE DOMAIN naziv_domena predefinisani domen  
predikat;
```

```
CREATE DOMAIN Kol INTEGER
```

```
∀ Kol (Kol > 5000 AND Kol < 50000 AND MOD (Kol, 50) =0);
```

ili

```
CREATE DOMAIN Boja CHAR(6) IN ('Crvena', 'Plava', 'Bela',  
'Zelena');
```

Ograničenje je zadovoljeno ako predikat dobije vrednost "T" (True- istinito).

# PRAVILA INTEGRITETA ZA ATRIBUTE

Očigledno je da se pravila integriteta za attribute mogu definisati preko sledeće četvorke, odnosno tabele:

<naziv atributa, domen, ograničenje, akcija>

# PRAVILA INTEGRITETA ZA RELACIJE

Sintaksa za iskazivanje ove vrste integriteta, bez navođenja trigger procedure, može da bude:

CREATE INTEGRITY RULE    naziv ograničenja predikat;

Promenljive u predikatu mogu da budu samo atributi jedne relacije. Atribut kao promenljiva se označava sa "dot notacijom". Primer pravila integriteta za relacije bi mogao biti:

CREATE INTEGRITY RULE Starost\_smer

$\forall$  Student (IF Student.ŠSmer = 01 THEN Student.Starost < 28);

# PRAVILA INTEGRITETA ZA BAZU

Preko ovog pravila moguće je iskazati bilo kakvo složeno ograničenje na vrednosti atributa u bazi podataka, ograničenje koje povezuje vrednosti atributa iz više relacija. Sintaksa za iskaz ove vrste ograničenja je jednaka onoj za ograničenja nad jednom relacijom.

Primer ove vrste ograničenja je:

```
CREATE INTEGRITY RULE Ocene_smer  
∀ Student (IF Student.ŠifSmer = 01 THEN  
∃ Prijava (Prijava.Ocena >7 AND  
Student.BrInd = Prijava.BrInd  
AND Prijava.ŠifPred = Predmet.ŠifPred  
AND Predmet.Nazivpred = 'Matematika'.));
```

# PRAVILA INTEGRITETA PRELAZA IZ STANJA U STANJE

Preko ovog pravila moguće je iskazati bilo kakvo složeno ograničenje na vrednosti atributa u bazi podataka, ograničenje koje povezuje vrednosti atributa iz više relacija. Sintaksa za iskaz ove vrste ograničenja je jednaka onoj za ograničenja nad jednom relacijom.

Na primer, pretpostavimo pravilo: "Student sa smerom sa ŠifSmer = 01 ne može se prebaciti na smer sa ŠifSmer = 05"

```
CREATE INTEGRITY RULE Prelaz sa smerom_na_smer  
∀ Student, Student' (IF Student'.ŠifSmer=01  
THEN Student.ŠifSmer ≠ 05);
```

# **OPERACIJE RELACIONOG MODELA**

**Relaciona algebra  
Relacioni račun**

# RELACIONA ALGEBRA

- ❑ Relaciona algebra definiše skup operacija pomoću kojih je moguće dobiti željenu relaciju (tabelu) iz skupa datih relacija (tabela).
  - ❑ **Konvencionalne skupovne operacije, unija, presek, razlika i Dekartov proizvod;**
  - ❑ **Specijalne relacione operacije selekcija, projekcija, spajanje i deljenje;**
  - ❑ **Dodatne operacije relacione algebre,** operacije koje su se kasnije dodavale originalnoj relacionoj algebri da bi se povećala njena "moć" kao upitnog jezika- Skalarna računanja u relacionoj algebri;
  - ❑ **Operacije ažuriranja baze;**
  - ❑ **Operacije u prisustvu nula vrednosti;**



# SKUPOVNE OPERACIJE

- **RELACIJA JE SKUP:**

- UNIJA, PRESEK, RAZLIKA I DEKARTOV PROIZVOD

- **Uslov kompatibilnosti** relacija  $R_1$  i  $R_2$  za izvođenje operacija unije, preseka i diferencije:

**Relacije  $R_1$  i  $R_2$  moraju imati isti broj atributa (isti stepen), a njihovi odgovarajući atributi moraju biti definisani nad istim domenima da bi se nad njima mogle izvršiti operacije unije, preseka i diferencije.**

# Tabele za primer

**S<sub>1</sub>**

BrInd	MLB	Ime	Starost	[ ifSmer
152/97	16309723331981	Ana	19	01
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02

**S<sub>2</sub>**

BrInd	MLB	Ime	Starost	[ ifSmer
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02
003/94	23456786418976	Milo{	22	01

# UNIJA

$$S_3 := S_1 \cup S_2$$

**S<sub>3</sub>**

BrInd	MLB	Ime	Starost	[ ifSmer
152/97	16309723331981	Ana	19	01
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02
003/94	23456786418976	Milo{	22	01

# RAZLIKA

$$S_4 := S_1 - S_2$$

**S<sub>4</sub>**

BrInd	MLB	Ime	Starost	[ ifSmer
152/97	16309723331981	A na	19	01

# PRESEK

$$S_5 := S_1 \cap S_2$$

**S<sub>5</sub>**

BrInd	MLB	Ime	Starost	[ ifSmer
223/95	13975673331981	Mirko	21	01
021/94	11145276418976	Zoran	20	02

# DEKARTOV PROIZVOD

$R_1$

C	D	E
1	2	3
2	4	6

$R_2$

A	B
a1	b1
a2	b2
a3	b3

$R_3 = R_1 \times R_2$

C	D	E	A	B
1	2	3	a1	b1
1	2	3	a2	b2
1	2	3	a3	b3
2	4	6	a1	b1
2	4	6	a2	b2
2	4	6	a3	b3

Nekontrolisano  
spajanje.

# SPECIJALNE RELACIONE OPERACIJE

- **Projekcija**
- **Selekcija**
- **Spajanje**
- **Deljenje**

# PROJEKCIJA

- Neka je  $R(A_1, A_2, \dots, A_n)$  relacija, a  $X$  podskup njenih atributa. Označimo sa  $Y$  komplement  $\{A_1, A_2, \dots, A_n\} - X$ . Rezultat operacije projekcije relacije  $R$  po atributima  $X$  je

$$\pi_X(R) = \{x \mid \exists y, \langle x, y \rangle \in R\}.$$



# Projekcija - primer

Građanin

MLB	Ime	Starost	MestoRo
16309723331981	A na	19	Beograd
13975673331981	Mirko	21	Valjevo
11145276418976	Zoran	20	Beograd
23243723331981	A na	19	Ni{
22222223331981	Mirko	21	Beograd
11145276418976	Zoran	20	Novi Sad
23456786418976	Milo{	22	Beograd

$Pr_1 := \pi_{\text{Ime, Starost}} \text{Građanin}$   
**Pr<sub>1</sub>**

Ime	Starost
A na	19
Mirko	21
Zoran	20
Milo{	22

# SELEKCIJA

Data je relacija  $R(A_1, A_2, \dots, A_n)$  i predikat  $\Theta$  definisan nad njenim atributima. Rezultat operacije selekcije

$$\sigma_{\Theta}(R) = \{x \mid x \in R \text{ AND } \Theta(x)\}$$

( skup n-torki  $x$  relacije  $R$  koje zadovoljavaju predikat (uslov)  $\Theta$  ).

## SELEKCIJA - PRIMER

Primer: Prikaži građane koji su stariji od 20 godina i rođeni su u Beogradu.

$Pr_2 := \sigma_{\text{Starost} > 20 \text{ AND MestoRođ} = \text{"Beograd"}} (\text{Građanin})$

**$Pr_2$**

MLB	Ime	Starost	MestoRođ
22222223331981	Mirko	21	Beograd
23456786418976	Milo{	22	Beograd

Restrikcija – koje n-torke mogu da se jave u rezultatu

## SPAJANJE - JOIN

Date su relacije  $R_1(A_1, A_2, \dots, A_n)$  i  $R_2(B_1, B_2, \dots, B_m)$  i predikat  $\Theta$  definisan nad njihovim atributima. Obeležimo sa  $X$  i  $Y$  skupove atributa relacija  $R_1$  i  $R_2$ , respektivno. Rezultat operacije spajanja ovih relacija (tzv. **teta spajanje**) je

$$R_1[x\Theta]R_2 = \{ \langle x, y \rangle \mid x \in R_1 \text{ AND } y \in R_2 \text{ AND } \Theta(x, y) \}.$$

Oznaka  $x\Theta$  za operaciju spajanja ukazuje na činjenicu, očiglednu iz definicije teta spajanja, da ova operacija nije primitivna operacija relacione algebre, već se može izvesti uzastopnom primenom operacije Dekartovog proizvoda ( $\times$ ) nad relacijama koje se spajaju i selekcije po predikatu  $\Theta$  nad tako dobijenom relacijom.

# SPAJANJE

- Ako je predikat  $\Theta$  definisan sa  $A_k = B_j$ , s tim da su i atributi  $A_k$  i  $B_j$  definisani nad istim domenima, tada se takvo spajanje naziva **ekvispajanje**.
- Očigledno je da se u rezultatu ekvispajanja uvek pojavljuju dve iste kolone, u gornjem primeru dve iste kolone MLB. Ako se jedna od te dve kolone izbace, takvo spajanje se naziva **prirodno spajanje**. Predpostavlja se da su atributi spajanja istoimeni.

# SPAJANJE - PRIMERI

## Gradanin

MLB	Ime	Starost	MestoRođ
16309723331981	Ana	19	Beograd
13975673331981	Mirko	21	Valjevo
11145276418976	Zoran	20	Beograd
23243723331981	Ana	19	Niš
22222223331981	Mirko	21	Beograd
11145276418976	Zoran	20	Novi Sad
23456786418976	Miloš	22	Beograd

## Student

BrInd	MLB	Smer
152/97	16309723331981	01
223/95	13975673331981	01
021/94	11145276418976	02
003/94	23456786418976	01

# SPAJANJE - PRIMERI

$Pr_3 := \text{Građanin} [\text{Građanin.MLB} = \text{Student.MLB}] \text{ Student}$

**Pr<sub>3</sub>**

MLB	Ime	Starost	MestoRo	MLB	BrInd	Smer
16309723331981	Ana	19	Beograd	16309723331981	152/87	01
13975673331981	Mirko	21	Valjevo	13975673331981	223/95	01
11145276418976	Zoran	20	Beograd	11145276418976	021/94	02
23456786418976	Milo{	22	Beograd	23456786418976	003/94	01

$Pr_4 := \text{Građanin} * \text{Student}$

**Pr<sub>4</sub>**

MLB	Ime	Starost	MestoRo	BrInd	Smer
16309723331981	Ana	19	Beograd	152/87	01
13975673331981	Mirko	21	Valjevo	223/95	01
11145276418976	Zoran	20	Beograd	021/94	02
23456786418976	Milo{	22	Beograd	003/94	01

# DELJENJE

1. Deljenje je operacija pogodna za upite u kojima se javlja reč "svi" ("sve", "sva"). Formalno se definiše na sledeći način:

Neka su  $A(X,Y)$  i  $B(Z)$  relacije gde su  $X$ ,  $Y$  i  $Z$  skupovi atributa takvi da su  $Y$  i  $Z$  jednakobrojni, a odgovarajući domeni su im jednaki. Rezultat operacije deljenja

$$A[Y \div Z]B = R(X)$$

gde  $n$ -torka  $x$  uzima vrednosti iz  $A.X$ , a par  $\langle x,y \rangle$  postoji u  $A$  za sve vrednosti  $y$  koje se pojavljuju u  $B(Z)$ .



# DELJENJE

## Prijava

Predmet		BrInd	ŠifPred
ŠifPred		152/97	P1
P1		152/97	P2
P2		021/94	P1
P3		003/94	P3
		152/97	P3

**Prijava [ Prijava.ŠifPred ÷ Predmet.ŠifPred] Predmet**

BrInd
152/97

# DELJENJE

operacija deljenja nije primitivna operacija relacionalne algebre, već se može izvesti pomoću drugih operacija na sledeći način:

$$(X, Y) [Y \div Z] B(Z) = \pi_X A - \pi_X ((\pi_X A \times B) - A)$$

objašnjenje:

- 

$\pi_X A$  daje sve n-torke koje mogu da učestvuju u rezultatu,

- 

$\pi_X A \times B$  daje relaciju u kojoj se za svaku vrednost  $z$  iz  $B$  pojavljuju parovi  $\langle x, z \rangle$  sa svim vrednostima  $x$ ,

-

# RELACIONA ALGEBRA

9. ***Poluspajanje (SEMIJOIN).*** Operacija poluspajanja uvedena je za potrebe distribuiranih baza podataka. Ako se, u nekoj distribuciji relacione baze podataka, relacije koje se spajaju nalaze na različitim lokacijama, spajanje je veoma skupa operacija, pogotovo ako se rezultat zahteva sa neke treće lokacije. Posmatrajmo sledeći primer:

Radnik(ŠifRadn, ImeRadn, AdresaRadn, Staž, ŠifOdel) -  
lokacija a

Odeljenje(ŠifOdel, NazivOdel) - lokacija b,

a na lokaciji c se izvršava operacija

Radnik \* Odeljenje

# RELACIONA ALGEBRA

## Skalarna računanja u relacionoj algebri

- **EXTEND** koji uključuje tzv "horizontalno računanje" u relacionu algebru, odnosno formira vrednost atributa relacije preko aritmetičkog izraza nad vrednostima ostalih atributa iste n-torke;
- **SUMMARIZE** koji uključuje tzv "vertikalno računanje" u relacionu algebru, odnosno formira novu relaciju kao projekciju polazne po nekim atributima, proširenu za atribut koji je rezultat nekog "vertikalnog", agregirajućeg, aritmetičkog izraza nad vrednostima odgovarajućih atributa.

# RELACIONA ALGEBRA

## Operacije sa nula vrednostima

- **MOŽDA\_SELEKCIJA (MAYBE\_SELECT).** Selektuju se one n-torke relacije za koje se predikat selekcije, na osnovu trovrednosnih tablica istinitosti, sračunava u nula vrednost.
- **SPOLJNO\_SPAJANJE (OUTER\_JOIN):**
  - centralno, levo i desno
- **SPOLJNA\_UNIJA (OUTER\_UNION).**

# RELACIONI RAČUN

- Postoje dva oblika relacionog računa:
  - (1) **Relacioni račun n-torki** i
  - (2) **Relacioni račun domena**
- **Relacioni račun n-torki** je Predikatski račun prvog reda u kome promenljive uzimaju vrednosti n-torki relacija date baze podataka.
- **Relacioni račun domena** je Predikatski račun prvog reda u kome promenljive uzimaju vrednosti iz nekih domena atributa relacija date baze podataka.

# PREDIKATSKI RAČUN PRVOG REDA

- (1) Afirmativna rečenica, koja ima smisla i koja je istinita ili neistinita naziva se sud.
- (2) Afirmativna rečenica koja ima smisla i koja sadrži jedan ili više promenljivih parametara i koja postaje sud uvek kada parametri iz rečenice dobiju konkretnu vrednost naziva se predikat. Broj parametara u predikatu se naziva dužina predikata. (Primer predikata je  $x^2 + y^2 \leq 1$ ).
- (3) Činjenica da u predikatskom računu promenljiva  $x$  uzima svoju vrednost iz nekog skupa  $R$  označava se sa  $x : R$ .
- (4) Predikatski ili kvantifikatorski račun je matematička teorija čiji su objekti formule koje predstavljaju predikate.
- (5) Definišu se **atomske formule** i **pravila (sintaksa) izvođenja složenijih formula**

# RELACIONI RAČUN N-TORKI

- U RELACIONOM RAČUNU N-TORKI:
  - Promenljive su n-torke relacija;
  - Atomske formule se definišu nad atributima n-torki;
  - Pravila izvođenja su standardna pravila Predikatskog računa prvog reda.
- U OBJEKTNOM RAČUNU:
  - Promenljive su objekti klasa.
  - Atomske formule se definišu nad atributima i operacijama objekata;
  - Pravila izvođenja su standardna pravila Predikatskog računa prvog reda.



# RELACIONI RAČUN N-TORKI

- Činjenica da u relacionom računu n-torki promenljiva  $x$  uzima kao svoju vrednost n-torku relacije  $R$  označava se sa  $x : R$ .
- Atomi u relacionom računu n-torki su:
  - **$x.A \oplus y.B$**  ( $x.A$  je vrednost atributa  $A$  relacije  $R_1$  za n-torku  $x$  koja iz ove relacije uzima vrednost. Na isti način  $y.B$  je vrednost atributa  $B$  neke druge relacije  $R_2$  iz koje promenljiva  $y$  uzima n-torke kao svoje vrednosti ( $x: R_1, y: R_2$ ). Pretpostavlja se da su atributi  $A$  i  $B$  definisani nad istim domenom, a  $\oplus$  je operacija poređenja definisana nad tim domenom.)
  - **$x.A \oplus c$**  gde su  $x, A$  i  $\oplus$  kao i u prethodnom stavu, a  $c$  je konstanta koja ima isti domen kao i atribut  $A$ .

# RELACIONI RAČUN N-TORKI

- Formule se formiraju od atoma preko sledećih pravila (sintakse):

- Atom je formula;
- Ako je  $P1$  formula, tada su formule i  $\text{NOT } P1$  i  $(P1)$ ;
- Ako su  $P1$  i  $P2$  formule tada su formule i  $P1 \text{ AND } P2$  i  $P1 \text{ OR } P2$ ;
- Ako je  $P1(s)$  formula koja sadrži neku slobodnu promenljivu  $s$  tada su i  $\exists s (P1(s))$  i  $\forall s (P1(s))$  takođe formule  
( $\exists$  - "postoji", egzistencijalni kvantifikator,  
 $\forall$  - "za svako", univerzalni kvantifikator).

# RELACIONI RAČUN N-TORKI

- **Opšti iskaz relacionog računa.** Neka su  $R_1, R_2, \dots, R_n$  relacije u nekoj bazi podataka. Neka su  $A, B, \dots, C$  atributi ovih relacija, respektivno i neka je  $F$  formula. Opšti izraz relacionog računa n-torki je tada:

$$\begin{array}{l} t: R_1, u: R_2, \dots, v: R_n \\ t.A, u.B, \dots, v.C \textbf{ WHERE } F \end{array}$$

(Prikaži vrednosti atributa  $A$  relacije  $R_1$ , atribut  $B$  relacije  $R_2, \dots$  i atribut  $C$  relacije  $R_n$ , za one  $n$ -torke koje zadovoljavaju uslov definisan formulom  $F$ ).

# PRIMERI

PRIMERI ĆE BITI PRIKAZANI NA SLEDEĆJEDNOSTAVNOJ  
ŠEMI RELACIONE BAZE PODATAKA:

STUDENT(BI,IME, STAROST,ŠSMER)  
PRIJAVA(BI,ŠP,OCENA)  
PREDMET(ŠP, NAZIP,BRČAS)  
SMER(ŠSMER,NAZIVS)

# PRIMERI RELACIONOG RAČUNA N-TORKI

(a) *Prikaži brojeve indeksa i imena studenata koji su stariji od 20 godina.*

x: STUDENT

x.BI, x.IME WHERE x.STAROST > 20;

(b) *Prikaži imena studenata koji studiraju smer "Informacioni sistemi".*

x: STUDENT, y: SMER

x.IME WHERE  $\exists y$  (y.ŠSMER = x.ŠSMER  
AND y.NAZIVS = 'InfSist');

## PRIMERI RELACINOG RAČUNA N-TORKI

*(c) Prikaži imena studenata koji su položili sve predmete.*

x: STUDENT, y: PRIJAVA, z: PREDMET

x.IME WHERE  $\forall z (\exists y (y.BI = x.BI \text{ AND } y.ŠP = z.ŠP ))$ ;

# INTERPRETACIJA SQL-a PREKO RELACIONE ALGEBRE I RELACIONOG RAČUNA

- SQL se može interpretirati kao korisniku prilagođena sintaksa i za relacionu algebru i za relacioni račun n-torki.
- U relacionoj algebri, opšti izraz preko koga se može dobiti željena tabela iz skupa datih tabela je **"projekcija po navedenim atributima, selekcije po datom uslovu, Dekartovog proizvoda potrebnih tabela"**, odnosno

$$\Pi [R1.A1, R2.A2, \dots, Rm.An] (\Sigma_{E \wedge [\Theta]} (R1 \times R2 \times \dots \times Rm))$$

# INTERPRETACIJA SQL-a PREKO RELACIONE ALGEBRE I RELACIONOG RAČUNA

Isti upit se u SQL-u može izraziti na sledeći način:

```
SELECT R1.A1, R2.A2, ..., Rm.An  
FROM R1,R2,..., Rm  
WHERE  $\Theta$ ;
```

Očigledne su sledeće ekvivalencije:

- Ključna reč **SELECT** odgovara operaciji projekcije ( $\Pi$ )
- Ključna reč **FROM** odgovara Dekartovom proizvodu;
- Ključna reč **WHERE** odgovara operaciji selekcije ( $\Sigma$ ) po uslovu  $\Theta$ .



# INTERPRETACIJA SQL-a PREKO RELACIONE ALGEBRE I RELACIONOG RAČUNA

Ekvivalencija relacionog računa n-torki i SQL-a, može se pokazati ako se prethodni SQL upit napiše u obliku:

```
SELECT x.A1, y.A2, ..., z.An  
      FROM R1 x, R2 y, ..., Rm z  
      WHERE  $\Theta$ ;
```

Interpretacija je sledeća:

- Ključna reč **FROM** služi za definisanje promenljivih
- Ključna reč **SELECT** definiše "desnu stranu" opšteg iskaza relacionog računa
- Ključna reč **WHERE** ima istu ulogu kao i u opštem iskazu relacionog računa - iza nje se navodi predikat (uslov) koji treba da bude zadovoljen.

# IMPLEMENTACIJA OBJEKTNOG RAČUNA - QUEL

Najpoznatiji upitni jezik koji predstavlja direktnu implementaciju relacionog računa n-torki je **QUEL (Query Language)**. Promenljive koje uzimaju za svoje vrednosti n-torke pojedinih relacija navode se iskazom:

**RANGE OF** naziv-promenljive **IS** naziv-relacije  
čime se iskazuje da neka promenljiva uzima za svoje vrednosti n-torke navedene relacije. Opšti oblik upita u QUEL-u je:

**RETRIEVE** lista-atributa **WHERE** predikat;

## PRIMERI UPITA U QUEL-U SU:

- (1) *Prikazati brojeve indeksa i imena radnika koji studiraju smer "Informacioni sistemi".*

```
RANGE OF x IS STUDENT
RANGE OF y IS SMER
  RETRIEVE (x. BI, x.IME )
  WHERE x.ŠIFSMER = y.ŠIFSMER
  AND y.NAZIVSMER = 'Infsist';
```

- (2) *Prikaži parove imena studenata iste starosti.*

```
RANGE OF x IS STUDENT
RANGE OF y IS STUDENT
  RETRIEVE (x.IME, y.IME, x.STAROST)
  WHERE x.STAROST = y.STAROST
```

## PRIMERI JEDNOSTAVNIH SQL UPITA

*Prikaži brojeve indeksa i imena studenata koji su stariji od 20 godina.*

```
SELECT STUDENT.BI, STUDENT.IME  
FROM STUDENT  
WHERE STUDENT.STAROST >20;
```

*Prikaži imena studenata koji studiraju smer "Informacioni sistemi".*

```
SELECT STUDENT.BI  
FROM STUDENT, SMER  
WHERE STUDENT.ŠSMER = SMER.ŠSMER AND  
SMER.NAZIVS= 'InfSist';
```

# RELACIONI RAČUN DOMENA

U relacionom računu domena promenljive uzimaju vrednosti iz nekih domena atributa relacija posmatrane relacione baze podataka. Ovde se, pored navedenih, definiše još jedna atomska formula, tzv. "uslov članstva" (membership condition

$$R(\text{term}, \text{term}, \dots)$$

gde je R ime neke relacije a svaki term ima oblik A: v, gde je A neki atribut relacije R, a v je ili promenljiva ili konstanta. Uslov članstva se sračunava u TRUE ako postoji n-torka u relaciji R koja ima zadate vrednosti navedenih atributa.

STUDENT (BI = '152/97' , ŠIFSMER =01 )

# RELACIONI RAČUN DOMENA

Opšti izraz relacionog računa domena je:

$x, y, \dots, z$  **WHERE**  $F$

gde su  $x, \dots, z$  promenljive a  $F$  je formula koja uključuje i uslov članstva.

(a) *Prikaži brojeve indeksa i imena studenata starijih od 20 godina*

$x, y$  WHERE  $\exists z > 25$  AND  
STUDENT (BRIND:  $x$ , IME:  $y$ , STAROST:  $z$ )

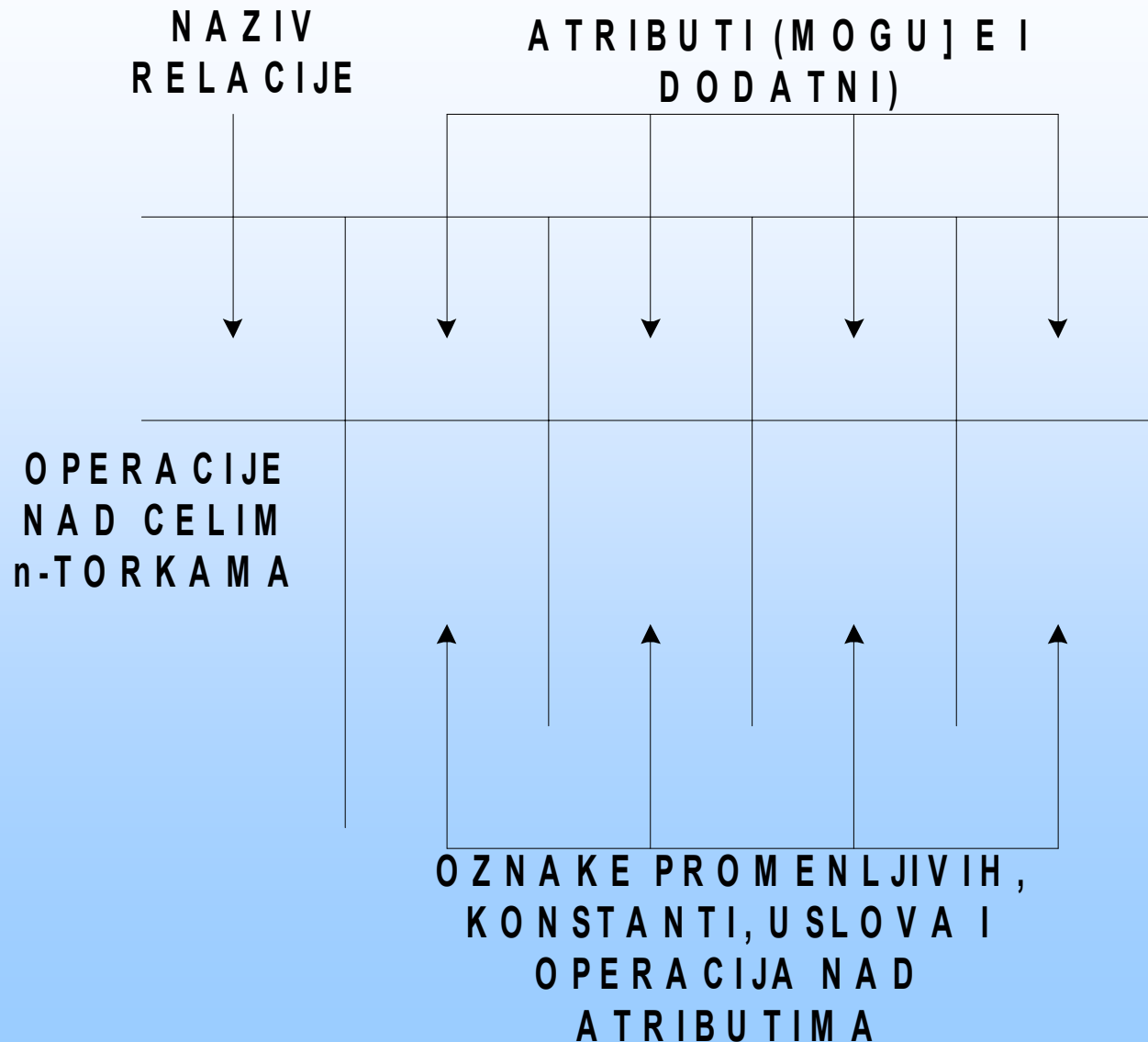
(b) *Prikaži imena studenata sa smerom "InfSis".*

$x$  WHERE  $\exists y$  ( (IME:  $x$ , ŠIFSMER:  $y$ )  
AND SMER (ŠIFSMER:  $y$ , NAZIVS: 'InfSis'))

# RELACIONI RAČUN DOMENA

- QBE (Query By Example) je upitni jezik koji predstavlja implementaciju relacionog računa domena, preko specifične "dvodimenzione" sintakse, koja je vrlo bliska korisniku, jer se preko nje, direktno, u skeletu tabele predstavljene na ekranu, zadaje "primer odgovora" koji korisnik želi. Otuda i ime Query By Example (upit na osnovu primera).

# QBE (Query By Example)





# PRIMER QBE

STUDENT	BI	IME	STAROST	[SMER
P.	_127/97			

PREDMET	[P	NAZIP	BR ^ A S
	_p01	'M atemat'	

PRIJAVA	BI	[P	OCENA
	_127/97	_p01	> 8