

SQL - Kratki pogled

- opisivanje relacijske sheme (kreiranje relacije)
 - kreira praznu relaciju
 - ujedno je moguće definirati i integritetska ograničenja

```
CREATE TABLE mjesto (
    pbr          INTEGER
, nazMjesto   CHAR (30)
, sifZup      SMALLINT
);
```



```
DROP TABLE mjesto;
```



Unijska kompatibilnost

- Dvije relacije su unijski kompatibilne ukoliko vrijedi:
 - relacije su istog stupnja
i
 - korespondentni atributi su definirani nad istim domenama

polozioMatem		
matBr	ime	prez
12345	Ivo	Kolar
13254	Ana	Horvat

polozioProgr		
mbr	prezSt	imeSt
92632	Ban	Jura
67234	Novak	Iva

- relacije su istog stupnja
- dom (matBr) = dom(mbr)
- dom (ime) = dom(imeSt)
- dom (prez) = dom(prezSt)

→ relacije su unijski kompatibilne

- kod ocjene jesu li relacije unijski kompatibilne
 - poredak atributa nije bitan
 - imena atributa nisu bitna

Skupovske operacije: unija, presjek, razlika

- Skupovske operacije (unija, presjek, razlika) mogu se obavljati isključivo nad UNIJSKI KOMPATIBILNIM relacijama

Unija

- Rezultat operacije $r_1 \cup r_2$ je relacija čije su n-torce elementi relacije r_1 ili elementi relacije r_2 ili elementi obje relacije.
 - n-torce koje su elementi obje relacije u rezultatu se pojavljuju samo jednom (jer relacija je SKUP n-torki)

polozioMatem

mbr	ime	prez
100	Ivan	Kolar
102	Ana	Novak
103	Tea	Ban
107	Jura	Horvat

polozioProgr

mbr	ime	prez
102	Ana	Novak
105	Rudi	Kolar
107	Jura	Horvat

polozioBaremJedan =
polozioMatem \cup polozioProgr

polozioBaremJedan

mbr	ime	prez
100	Ivan	Kolar
102	Ana	Novak
103	Tea	Ban
105	Rudi	Kolar
107	Jura	Horvat

studenti koji su položili ili
Matematiku ili
Programiranje ili
oba predmeta

$$r_1 \cup r_2 \equiv r_2 \cup r_1$$

Presjek

- Rezultat operacije $r_1 \cap r_2$ je relacija čije su n-torce elementi relacije r_1 i elementi relacije r_2

polozioMatem

mbr	ime	prez
100	Ivan	Kolar
102	Ana	Novak
103	Tea	Ban
107	Jura	Horvat

polozioProgr

mbr	ime	prez
102	Ana	Novak
105	Rudi	Kolar
107	Jura	Horvat

polozioOba =
polozioMatem \cap polozioProgr

polozioOba

mbr	ime	prez
102	Ana	Novak
107	Jura	Horvat

studenti koji su
položili i Matematiku
i Programiranje

$$r_1 \cap r_2 \equiv r_2 \cap r_1$$

Razlika

- Rezultat operacije $r_1 \setminus r_2$ je relacija čije su n-torce elementi relacije r_1 i nisu elementi relacije r_2

polozioMatem

mbr	ime	prez
100	Ivan	Kolar
102	Ana	Novak
103	Tea	Ban
107	Jura	Horvat

polozioProgr

mbr	ime	prez
102	Ana	Novak
105	Rudi	Kolar
107	Jura	Horvat

**polozioSamoMatem =
polozioMatem \ polozioProgr**

polozioSamoMatem

mbr	ime	prez
100	Ivan	Kolar
103	Tea	Ban

studenti koji su
položili Matematiku,
ali **nisu** položili
Programiranje

$$r_1 \setminus r_2 \neq r_2 \setminus r_1$$

polozioSamoProgr = polozioProgr \ polozioMatem

polozioSamoProgr

mbr	ime	prez
105	Rudi	Kolar

SQL - Projekcija

- za ispravno obavljanje projekcije nije dovoljno u listi za selekciju samo navesti imena atributa prema kojima se obavlja projekcija:

- primjer koji ujedno pokazuje kako rezultat SQL naredbe ne mora uvijek biti relacija



```
SELECT tenor  
      , grad  
  FROM nastup;
```

Neispravna projekcija

tenor	grad
P. Domingo	London
P. Domingo	New York
P. Domingo	London
J. Carreras	New York
L. Pavarotti	Sydney
L. Pavarotti	London
L. Pavarotti	Sydney
L. Pavarotti	London

$\pi_{\text{tenor}, \text{grad}}(\text{nastup})$

```
SELECT DISTINCT tenor  
      , grad  
  FROM nastup;
```

Ispravna projekcija

tenor	grad
P. Domingo	London
P. Domingo	New York
J. Carreras	New York
L. Pavarotti	Sydney
L. Pavarotti	London

SQL - Selekcija

student	matBr	ime	prez	postBr
	100	Ivan	Kolar	52000
	102	Ana	Horvat	10000
	105	Jura	Novak	21000
	107	Ana	Ban	51000

$\sigma_{ime = 'Ana' \vee postBr > 31000} (student)$

```
SELECT * FROM student  
WHERE ime = 'Ana'  
OR postBr > 31000;
```



	matBr	ime	prez	postBr
	100	Ivan	Kolar	52000
	102	Ana	Horvat	10000
	107	Ana	Ban	51000

Kartezijev produkt

- Zadana je relacija $r(R)$ i relacija $s(S)$, pri čemu je $R \cap S = \emptyset$.
- Obavljanjem operacije $r \times s$ dobiva se relacija $p(P)$, $P = R \cup S$.
n-torce relacije p se dobivaju spajanjem (ulančavanjem) svake n-torce iz relacije r sa svakom n-torkom iz relacije s
 - $\deg(p) = \deg(r) + \deg(s)$
 - $\text{card}(p) = \text{card}(r) \cdot \text{card}(s)$

SQL - Preimenovanje atributa

- Primjer u kojem je potrebno koristiti preimenovanje atributa
 - SQL naredba bi bila ispravna i bez preimenovanja, ali tada kao rezultat ne bismo dobili relaciju (jer bi u shemi rezultata postojala dva atributa istog imena)

r	A	B
1	a	
2	b	
3	c	

s	B	C
c		α
d		β

$r \times \rho_{s(B2, C)}(s)$

```
SELECT A, r.B, s.B AS B2, C  
FROM r, s;
```

A	B	B2	C
1	a	c	α
1	a	d	β
2	b	c	α
2	b	d	β
3	c	c	α
3	c	d	β

SQL - Spajanje uz uvjet

- Koristi se ekvivalencija

$$r \triangleright\triangleleft_F s = \sigma_F(r \times s)$$

linija $\triangleright\triangleleft$ zrakoplov
dolet \geq udaljenost

linija		zrakoplov	
let	udaljenost	tip	dolet
CA-825	700	B747	13000
LH-412	4800	A320	5400
BA-722	15000	DC-9	3100
CA-311	13000		

```
SELECT *
  FROM linija, zrakoplov
 WHERE dolet >= udaljenost;
```

Kartezijev produkt

Selekcija

Linije i zrakoplovi koji na
tim linijama mogu letjeti

let	udaljenost	tip	dolet
CA-825	700	B747	13000
CA-825	700	A320	5400
CA-825	700	DC-9	3100
LH-412	4800	B747	13000
LH-412	4800	A320	5400
CA-311	13000	B747	13000

SQL - Spajanje uz uvjet

- drugačija sintaksa:
- ```
SELECT SELECT List
 FROM table [JOIN table ON joinCondition] ...
 [WHERE Condition]
```

```
SELECT *
 FROM linija JOIN zrakoplov
 ON dolet >= udaljenost;
```

- Spajanje uz uvjet triju relacija:

```
SELECT *
 FROM r1
 JOIN r2
 ON joinCondition
 JOIN r3
 ON joinCondition;
```

# SQL - Spajanje uz uvjet i selekcija

- Kako pronaći linije i zrakoplove koji na tim linijama mogu letjeti, ali samo za one linije na kojima je udaljenost veća od 4000 km

$\sigma_{\text{udaljenost} > 4000}(\text{linija} \bowtie \text{zrakoplov})$   
dolet  $\geq$  udaljenost

```
SELECT *
 FROM linija, zrakoplov
 WHERE dolet >= udaljenost
 AND udaljenost > 4000;
```

```
SELECT *
 FROM linija
 JOIN zrakoplov
 ON dolet >= udaljenost
 WHERE udaljenost > 4000;
```

| let    | udaljenost | tip  | dolet |
|--------|------------|------|-------|
| LH-412 | 4800       | B747 | 13000 |
| LH-412 | 4800       | A320 | 5400  |
| CA-311 | 13000      | B747 | 13000 |

ili

# SQL - Spajanje uz uvjet i projekcija

- Kako pronaći tipove zrakoplova koji se mogu iskoristiti za letove na postojećim linijama

$$\pi_{\text{tip}}(\text{linija} \bowtie \text{zrakoplov})$$

dolet \geq \text{udaljenost}

```
SELECT DISTINCT tip
 FROM linija, zrakoplov
 WHERE dolet >= udaljenost;
```

| tip  |
|------|
| B747 |
| A320 |
| DC-9 |

ili

```
SELECT DISTINCT tip
 FROM linija
 JOIN zrakoplov
 ON dolet >= udaljenost;
```

# Spajanje s izjednačavanjem (*Equi-join*)

- Spajanje relacija s izjednačavanjem je poseban oblik spajanja uz uvjet u kojem se kao  $\theta$  operator koristi isključivo operator jednakosti (=)

| mjesto | pbr   | nazMjesto | sifZup |
|--------|-------|-----------|--------|
|        | 42000 | Varaždin  | 7      |
|        | 52100 | Pula      | 4      |
|        | 42230 | Ludbreg   | 7      |

| zupanija | sifZupanija | nazZup      |
|----------|-------------|-------------|
|          | 7           | Varaždinska |
|          | 4           | Istarska    |

$mjestouZupaniji = mjesto \triangleright\triangleleft zupanija$   
 $sifZup = sifZupanija$

| mjestouZupaniji | pbr   | nazMjesto | sifZup | sifZupanija | nazZup      |
|-----------------|-------|-----------|--------|-------------|-------------|
|                 | 42000 | Varaždin  | 7      | 7           | Varaždinska |
|                 | 52100 | Pula      | 4      | 4           | Istarska    |
|                 | 42230 | Ludbreg   | 7      | 7           | Varaždinska |

- Problem spajanja s izjednačavanjem relacija  $r(R)$  i  $s(S)$  kod kojih je  $R \cap S \neq \emptyset$ , rješava se na jednak način kao kod Kartezijskog produkta (korištenjem operatora preimenovanja)

# SQL - Spajanje s izjednačavanjem

- Koristi se ekvivalencija

$$r \triangleright\triangleleft s = \underset{F}{\sigma}_F(r \times s)$$

mjesto  $\triangleright\triangleleft$  zupanija  
sifZup = sifZupanija

```
SELECT *
 FROM mjesto, zupanija
 WHERE sifZup = sifZupanija;
```

| mjesto | pbr      | nazMjesto | sifZup |
|--------|----------|-----------|--------|
| 42000  | Varaždin | 7         |        |
| 52100  | Pula     | 4         |        |
| 42230  | Ludbreg  | 7         |        |

| zupanija | sifZupanija | nazZup |
|----------|-------------|--------|
| 7        | Varaždinska |        |
| 4        | Istarska    |        |

ili

```
SELECT *
 FROM mjesto
 JOIN zupanija
 ON sifZup = sifZupanija;
```

# SQL - Spajanje s izjednačavanjem

- U slučaju kada u relacijama postoje istoimeni atributi

| mjesto | pbr      | nazMjesto | sifZup |
|--------|----------|-----------|--------|
| 42000  | Varaždin | 7         |        |
| 52100  | Pula     | 4         |        |
| 42230  | Ludbreg  | 7         |        |

| zupanija | sifZup | nazZup      |
|----------|--------|-------------|
|          | 7      | Varaždinska |
|          | 4      | Istarska    |

$$\text{mjesto} \underset{\text{sifZup} = \text{sifZup2}}{\triangleright\triangleleft} \rho_{\text{zupanija}(\text{sifZup2}, \text{nazZup})} \text{zupanija}$$

Za razliku od relacijske algebre, u SQL-u nije nužno preimenovati atribut prije spajanja:

```
SELECT mjesto.*
 , zupanija.sifZup AS sifZup2
 , zupanija.nazZup
 FROM mjesto, zupanija
 WHERE mjesto.sifZup = zupanija.sifZup;
```

slično i u slučaju korištenja drugačije sintakse (ANSI join)

# SQL - Prirodno spajanje

- prirodno spajanje se razlikuje od spajanja s izjednačavanjem po tome što se istoimeni atributi iz dviju relacija izbacuju (tako da od svakog ostane samo po jedan)

| mjesto | pbr   | nazMjesto | sifZup |
|--------|-------|-----------|--------|
|        | 42000 | Varaždin  | 7      |
|        | 52100 | Pula      | 4      |
|        | 42230 | Ludbreg   | 7      |

| zupanija | sifZup | nazZup      |
|----------|--------|-------------|
|          | 7      | Varaždinska |
|          | 4      | Istarska    |

```
SELECT mjesto.* , zupanija.nazZup
FROM mjesto, zupanija
WHERE mjesto.sifZup = zupanija.sifZup;
```

| pbr   | nazMjesto | sifZup | nazZup      |
|-------|-----------|--------|-------------|
| 42000 | Varaždin  | 7      | Varaždinska |
| 52100 | Pula      | 4      | Istarska    |
| 42230 | Ludbreg   | 7      | Varaždinska |

# SQL - Prirodno spajanje

- drugačija sintaksa:

```
SELECT mjesto.* , zupanija.nazZup
FROM mjesto JOIN zupanija
 ON mjesto.sifZup = zupanija.sifZup;
```

# Agregacija

| ispit | mbrStud | akGod | nazPred      | ocjena |
|-------|---------|-------|--------------|--------|
|       | 100     | 2005  | Matematika   | 3      |
|       | 101     | 2005  | Matematika   | 5      |
|       | 102     | 2005  | Matematika   | 2      |
|       | 103     | 2006  | Matematika   | 3      |
|       | 100     | 2004  | Fizika       | 5      |
|       | 101     | 2006  | Fizika       | 5      |
|       | 102     | 2006  | Fizika       | 2      |
|       | 100     | 2005  | Vjerojatnost | 4      |

- Prosjek ocjena na svim ispitima (rješenje):

$$\rho_{\text{projek}(\text{projekOcj})}(G_{\text{AVG}(\text{ocjena})}(\text{ispit}))$$

projek

projekOcj  
3.625

```
SELECT AVG(ocjena) AS projekOcj
FROM ispit;
```

projekOcj  
3.625

# Agregacija (primjeri ostalih agregatnih funkcija)

| osoba |        |        |
|-------|--------|--------|
| sifra | tezina | visina |
| 101   | 62     | 170    |
| 103   | 94     | 186    |
| 105   | 74     | 181    |
| 107   | 62     | 165    |

$\rho_{rez1(broj1)}(G_{COUNT(sifra)}(osoba))$

| rez1 | broj1 |
|------|-------|
|      | 4     |

$\rho_{rez2(broj2)}(G_{SUM(tezina)}(osoba))$

| rez2 | broj2 |
|------|-------|
|      | 292   |

$\rho_{rez3(broj3)}(G_{AVG(visina)}(osoba))$

| rez3 | broj3 |
|------|-------|
|      | 175.5 |

$\rho_{rez4(broj4)}(G_{MAX(visina)}(osoba))$

| rez4 | broj4 |
|------|-------|
|      | 186   |

$\rho_{rez5(broj5)}(G_{MIN(tezina)}(osoba))$

| rez5 | broj5 |
|------|-------|
|      | 62    |

Moguće je odjednom izračunati više agregatnih vrijednosti:

$\rho_{rez6(broj6, broj7, broj8)}(G_{MIN(tezina), AVG(visina), MAX(visina)}(osoba))$

| rez6 | broj6 | broj7 | broj8 |
|------|-------|-------|-------|
|      | 62    | 175.5 | 186   |

# SQL - Agregatne funkcije

- naziv rezultantnog atributa nije definiran operacijom, stoga se koristi AS operator za preimenovanje

| osoba | sifra | tezina | visina |
|-------|-------|--------|--------|
|       | 101   | 62     | 170    |
|       | 103   | 94     | 186    |
|       | 105   | 74     | 181    |
|       | 107   | 62     | 165    |

```
SELECT COUNT(sifra) AS broj1 FROM osoba;
```

broj1  
4

```
SELECT SUM(tezina) AS broj2 FROM osoba;
```

broj2  
292

```
SELECT AVG(visina) AS broj3 FROM osoba;
```

broj3  
175.5

```
SELECT MAX(visina) AS broj4,
 MIN(tezina) AS broj5
FROM osoba;
```

|       |       |
|-------|-------|
| broj4 | broj5 |
| 186   | 62    |

# Agregacija i grupiranje

| ispit | mbrStud | akGod | nazPred      | ocjena |
|-------|---------|-------|--------------|--------|
|       | 100     | 2005  | Matematika   | 3      |
|       | 101     | 2005  | Matematika   | 5      |
|       | 102     | 2005  | Matematika   | 2      |
|       | 103     | 2006  | Matematika   | 3      |
|       | 100     | 2004  | Fizika       | 5      |
|       | 101     | 2006  | Fizika       | 5      |
|       | 102     | 2006  | Fizika       | 2      |
|       | 100     | 2005  | Vjerojatnost | 4      |

}

- Za svaki predmet ispisati prosječnu ocjenu (ispravno rješenje):

$\rho_{\text{projek}(nazPred, \text{projOcj})}(\text{nazPred} G_{\text{AVG}(ocjena)}(\text{ispit}))$

- grupirati po nazPred
- za svaku grupu izračunati AVG(ocjena)
- za svaku grupu formirati po jednu n-torku s vrijednošću atributa nazPred i izračunatim projekom
- obaviti operaciju preimenovanja

| projek       |         |
|--------------|---------|
| nazPred      | projOcj |
| Matematika   | 3.25    |
| Fizika       | 4       |
| Vjerojatnost | 4       |

# SQL - Grupiranje

- **SELECT *SELECT List***  
**FROM ...**  
**[WHERE *Condition*]**  
**[GROUP BY *column* [, *column*] ...]**

$\rho_{\text{projek1}}(\text{nazPred}, \text{akGod}, \text{projOcj}, \text{maxOcj}) \left( \text{nazPred}, \text{akGod}, \text{AVG(ocjena)}, \text{MAX(ocjena)} \right) \text{(ispit)}$

```
SELECT nazPred
 , akGod
 , AVG(ocjena) AS projOcj
 , MAX(ocjena) AS maxOcj
 FROM ispit
 GROUP BY nazPred, akGod;
```

| nazPred      | akGod | projOcj | maxOcj |
|--------------|-------|---------|--------|
| Matematika   | 2005  | 3.333   | 5      |
| Matematika   | 2006  | 3       | 3      |
| Fizika       | 2004  | 5       | 5      |
| Fizika       | 2006  | 3.5     | 5      |
| Vjerojatnost | 2005  | 4       | 4      |

# SQL - Grupiranje

- svi atributi koji se nalaze u listi za selekciju, a koji nisu argumenti agregatnih funkcija, moraju biti navedeni u GROUP BY dijelu naredbe

```
SELECT nazPred
 , akGod
 , mbrStud
 , AVG(ocjena) AS prosjOcj
 , MAX(ocjena) AS maxOcj
 FROM ispit
 GROUP BY nazPred, akGod;
```

**NEISPRAVNO!**

←

ispit

| mbrStud | akGod | nazPred      | ocjena |
|---------|-------|--------------|--------|
| 100     | 2005  | Matematika   | 3      |
| 101     | 2005  | Matematika   | 5      |
| 102     | 2005  | Matematika   | 2      |
| 103     | 2006  | Matematika   | 3      |
| 100     | 2004  | Fizika       | 5      |
| 101     | 2006  | Fizika       | 5      |
| 102     | 2006  | Fizika       | 2      |
| 100     | 2005  | Vjerojatnost | 4      |

Zašto je to neispravno?  
Za svaku grupu se formira samo po jedna n-torka: što s onim grupama u kojima postoji više vrijednosti atributa mbrStud?

| nazPred      | akGod | mbrStud         | prosjOcj | maxOcj |
|--------------|-------|-----------------|----------|--------|
| Matematika   | 2005  | 100, 101, 102 ? | 3.333    | 5      |
| Matematika   | 2006  | ?               | 3        | 3      |
| Fizika       | 2004  | ?               | 5        | 5      |
| Fizika       | 2006  | ?               | 3.5      | 5      |
| Vjerojatnost | 2005  | ?               | 4        | 4      |

# NULL vrijednost u izrazima

- Neka je binarni operator  $\alpha \in \{ +, -, *, / \}$ , a X i Y su izrazi
  - ako jedan ili oba operanda X, Y poprimaju NULL vrijednost, tada je rezultat izraza X  $\alpha$  Y također NULL vrijednost

5 + NULL → NULL

NULL - NULL → NULL

NULL \* 0 → NULL

- Neka je unarni operator  $\beta \in \{ +, - \}$ , a X je izraz
  - ako operand X poprima NULL vrijednost, tada je rezultat izraza  $\beta$  X također NULL vrijednost

- NULL → NULL

# NULL vrijednost u uvjetima usporedbe

- Neka su  $X$  i  $Y$  izrazi, a  $\gamma$  je operator usporedbe
$$\gamma \in \{ <, \leq, =, \neq, >, \geq \}$$
- ako niti jedan od operanada  $X$ ,  $Y$  nije NULL vrijednost, tada je rezultat izraza  $X \gamma Y$  logička vrijednost istina (*true*) ili logička vrijednost laž (*false*)
- ako jedan ili oba operanda  $X$ ,  $Y$  jesu NULL vrijednosti, tada je rezultat izraza  $X \gamma Y$  logička vrijednost nepoznato (*unknown*)

|                                |                       |
|--------------------------------|-----------------------|
| $7 \geq 5$                     | $\rightarrow$ true    |
| 'atlas' > 'zvuk'               | $\rightarrow$ false   |
| $-17.8 \leq \text{NULL}$       | $\rightarrow$ unknown |
| $\text{NULL} = \text{NULL}$    | $\rightarrow$ unknown |
| $\text{NULL} \neq \text{NULL}$ | $\rightarrow$ unknown |

# Operacija selekcije - NULL vrijednosti

- Obavljanjem **operacije selekcije**  $\sigma_F(r)$  dobiva se relacija koja sadrži samo one n-torke relacije r za koje je vrijednost predikata F istina (*true*). To znači da se n-torke za koje je vrijednost predikata F laž (*false*) ili nepoznato (*unknown*) ne pojavljuju u rezultatu

| r | A    | B |
|---|------|---|
| 1 | 20   |   |
| 2 | NULL |   |
| 3 | 60   |   |

$$s = \sigma_{B \leq 50}(r)$$

$20 \leq 50 \rightarrow \text{true}$   
 $\text{NULL} \leq 50 \rightarrow \text{unknown}$   
 $60 \leq 50 \rightarrow \text{false}$

| s | A  | B |
|---|----|---|
| 1 | 20 |   |

| r | A    | B |
|---|------|---|
| 1 | 20   |   |
| 2 | NULL |   |
| 3 | 60   |   |

$$s = \sigma_{B \neq 20}(r)$$

$20 \neq 20 \rightarrow \text{false}$   
 $\text{NULL} \neq 20 \rightarrow \text{unknown}$   
 $60 \neq 20 \rightarrow \text{true}$

| s | A  | B |
|---|----|---|
| 3 | 60 |   |

# SQL - operatori usporedbe IS NULL, IS NOT NULL

| student | matBr | prez   | postBr |
|---------|-------|--------|--------|
|         | 100   | Kolar  | 52000  |
|         | 102   | Horvat | 10000  |
|         | 105   | Novak  | NULL   |
|         | 107   | Ban    | 10000  |

- U SQL-u nije dopušteno operatore usporedbe  $<$ ,  $\leq$ ,  $=$ ,  $\neq$ ,  $>$ ,  $\geq$  koristiti u kombinaciji s "konstantom" NULL (npr.  $=NULL$ ,  $\neq NULL$ , ...)

```
SELECT * FROM student
WHERE postBr = NULL;
```

**Neispravna naredba**

```
SELECT * FROM student
WHERE postBr IS NULL;
```



| matBr | prez  | postBr |
|-------|-------|--------|
| 105   | Novak | NULL   |

```
SELECT * FROM student
WHERE postBr IS NOT NULL;
```



| matBr | prez   | postBr |
|-------|--------|--------|
| 100   | Kolar  | 52000  |
| 102   | Horvat | 10000  |
| 107   | Ban    | 10000  |

- Rezultat logičkog izraza  $X IS NULL$  ili logičkog izraza  $X IS NOT NULL$  je uvijek ili *true* ili *false*

# Kopija n-torke

- Primjer:

| osoba |      |        |        | student |      |        |        |
|-------|------|--------|--------|---------|------|--------|--------|
| mbr   | ime  | prez   | postBr | mbr     | ime  | prez   | postBr |
| 100   | Ivan | Novak  | 10000  | 100     | Ivan | Novak  | NULL   |
| 102   | Ana  | Horvat | 21000  | 102     | Ana  | Horvat | 21000  |
| 103   | Tea  | Ban    | 52000  | 103     | Tea  | Ban    | 21000  |
| 105   | NULL | Kolar  | NULL   | 105     | NULL | Kolar  | NULL   |

Diagram illustrating the copying of rows from the 'osoba' table to the 'student' table. Red arrows labeled 'nije kopija' point from the 'osoba' table to the 'student' table for rows 100, 103, and 105. Green arrows labeled 'jest kopija' point from the 'osoba' table to the 'student' table for rows 100, 102, and 103.

# Agregacija - NULL vrijednosti

| ispit | mbrStud | nazPred    | ocjena |
|-------|---------|------------|--------|
|       | 100     | Matematika | NULL   |
|       | 101     | Matematika | 4      |
|       | 102     | Matematika | 3      |
|       | 103     | Matematika | 3      |
|       | 100     | Fizika     | NULL   |
|       | 101     | Fizika     | 3      |

```
SELECT COUNT(mbrStud) AS broj1
FROM ispit;
```

broj1  
6

```
SELECT COUNT(ocjena) AS broj2
FROM ispit;
```

broj2  
4

```
SELECT COUNT(ocjena) AS broj3 FROM ispit
WHERE mbrStud = 100;
```

broj3  
0

```
SELECT COUNT(ocjena) AS broj4 FROM ispit
WHERE mbrStud = 200;
```

broj4  
0

```
SELECT AVG(ocjena) AS broj5 FROM ispit;
```

broj5  
3.25

```
SELECT AVG(ocjena) AS broj6 FROM ispit
WHERE mbrStud = 100;
```

broj6  
NULL

```
SELECT AVG(ocjena) AS broj7 FROM ispit
WHERE mbrStud = 200;
```

broj7  
NULL

# Agregatna funkcija COUNT(\*)

- Agregatna funkcija COUNT(*imeAtributa*)
  - broji n-torke u kojima vrijednost atributa *imeAtributa* nije NULL vrijednost
- Agregatna funkcija COUNT(\*)
  - broji n-torke zanemarujući njihov sadržaj

| ispit | mbrStud | nazPred    | ocjena |
|-------|---------|------------|--------|
|       | 100     | Matematika | NULL   |
|       | 101     | Matematika | 4      |
|       | 102     | Matematika | 3      |
|       | 103     | Matematika | 3      |
|       | 100     | Fizika     | NULL   |
|       | 101     | Fizika     | 3      |

```
SELECT COUNT(ocjena) AS brojOcj
 , COUNT(*) AS brojRedaka
 FROM ispit;
```

| brojOcj | brojRedaka |
|---------|------------|
| 4       | 6          |

- Ne postoji agregatna funkcija COUNT(DISTINCT \*)

# Grupiranje - NULL vrijednosti

- pri obavljanju operacije grupiranja, grupiranje n-torki se obavlja tako da se vodi računa o definiciji kopije n-torki
  - ako se grupiranje obavlja prema atributima iz skupa X, tada u istu grupu ulaze one n-torke čije su X-vrijednosti međusobne kopije

```
SELECT akGod, nazPred, AVG(ocjena) AS prosj
 FROM ispit
 GROUP BY akGod, nazPred;
```

ispit

|                | mbrStud | akGod | nazPred | ocjena |
|----------------|---------|-------|---------|--------|
| t <sub>1</sub> | 100     | 2005  | NULL    | 3      |
| t <sub>2</sub> | 101     | NULL  | NULL    | 5      |
| t <sub>3</sub> | 102     | 2005  | NULL    | 2      |
| t <sub>4</sub> | 103     | 2006  | Fizika  | 3      |
| t <sub>5</sub> | 100     | NULL  | NULL    | 5      |
| t <sub>6</sub> | 101     | 2006  | Fizika  | 5      |
| t <sub>7</sub> | 102     | 2005  | NULL    | 2      |

X = { akGod, nazPred }

t<sub>1</sub>(X), t<sub>3</sub>(X) i t<sub>7</sub>(X) su međusobne kopije

t<sub>2</sub>(X) i t<sub>5</sub>(X) su međusobne kopije

t<sub>4</sub>(X) i t<sub>6</sub>(X) su međusobne kopije

| akGod | nazPred | prosj  |
|-------|---------|--------|
| 2005  | NULL    | 2.3333 |
| NULL  | NULL    | 5      |
| 2006  | Fizika  | 4      |

# Lijevo vanjsko spajanje (*Left outer join*)

- sve n-torce relacije **student** će se pojaviti u rezultatu spajanja ako se primjeni operacija **lijevog vanjskog spajanja**

| student | matBr  | prez |
|---------|--------|------|
| 101     | Kolar  |      |
| 102     | Horvat |      |
| 103     | Novak  |      |

| upisanPred | matBrSt       | nazPred |
|------------|---------------|---------|
| 101        | Matematika    |         |
| 101        | Fizika        |         |
| 101        | Programiranje |         |
| 102        | Fizika        |         |

upisano = student \* $\triangleright\triangleleft$  upisanPred

matBr = matBrSt

| upisano |        | matBrSt | nazPred       |
|---------|--------|---------|---------------|
| matBr   | prez   |         |               |
| 101     | Kolar  | 101     | Matematika    |
| 101     | Kolar  | 101     | Fizika        |
| 101     | Kolar  | 101     | Programiranje |
| 102     | Horvat | 102     | Fizika        |
| 103     | Novak  | NULL    | NULL          |

- n-torkama "lijeve" relacije za koje ne postoje odgovarajuće n-torce u "desnoj" relaciji se kao vrijednosti atributa iz "desne" relacije postavljaju NULL vrijednosti

# SQL - Desno vanjsko spajanje (*Right outer join*)

student

| mbrSt | prezSt | temaSt        |
|-------|--------|---------------|
| 101   | Horvat | Tranzistori   |
| 103   | Novak  | Teslini izumi |
| 105   | Kolar  | Teorija kaosa |

nastavnik

| sifNast | prezNast | temaNast      |
|---------|----------|---------------|
| 202     | Ban      | Teslini izumi |
| 204     | Toplek   | Elektrane     |
| 206     | Oreb     | Teslini izumi |
| 209     | Pernar   | Teorija kaosa |

student  $\triangleright\triangleleft^*$  nastavnik  
temaSt = temaNast

```
SELECT student.* , nastavnik.* ←
FROM student RIGHT OUTER JOIN nastavnik
ON temaSt = temaNast;
```

ili SELECT \*

| mbrSt | prezSt | temaSt        | sifNast | prezNast | temaNast      |
|-------|--------|---------------|---------|----------|---------------|
| 103   | Novak  | Teslini izumi | 202     | Ban      | Teslini izumi |
| NULL  | NULL   | NULL          | 204     | Toplek   | Elektrane     |
| 103   | Novak  | Teslini izumi | 206     | Oreb     | Teslini izumi |
| 105   | Kolar  | Teorija kaosa | 209     | Pernar   | Teorija kaosa |

# SQL - Puno vanjsko spajanje (*Full outer join*)

student

| mbrSt | prezSt | temaSt        |
|-------|--------|---------------|
| 101   | Horvat | Tranzistori   |
| 103   | Novak  | Teslini izumi |
| 105   | Kolar  | Teorija kaosa |

nastavnik

| sifNast | prezNast | temaNast      |
|---------|----------|---------------|
| 202     | Ban      | Teslini izumi |
| 204     | Toplek   | Elektrane     |
| 206     | Oreb     | Teslini izumi |
| 209     | Pernar   | Teorija kaosa |

student \*  $\triangleright\triangleleft$  nastavnik

temaSt = temaNast

```
SELECT student.* , nastavnik.*
FROM student FULL OUTER JOIN nastavnik
ON temaSt = temaNast;
```

ili SELECT \*

# SQL - Tipovi podataka

## ■ DATE

- podaci ovog tipa se uvijek prikazuju u obliku datuma (npr. 18.11.2006). Interno je podatak predstavljen brojem dana proteklih od 31.12.1899. Ovaj tip podatka omogućava korištenje sljedećih operacija zbrajanja i oduzimanja:

- $\text{dat1} - \text{dat2}$
- $\text{dat} + \text{cijeliBroj}$
- $\text{dat} - \text{cijeliBroj}$

rezultat je podatak tipa INTEGER - broj dana proteklih između  $\text{dat2}$  i  $\text{dat1}$   
rezultat je podatak tipa DATE - izračunava koji datum je  $\text{cijeliBroj}$  dana nakon dana  $\text{dat}$   
rezultat je podatak tipa DATE - izračunava koji datum je  $\text{cijeliBroj}$  dana prije dana  $\text{dat}$

Konstante:

'17.2.2007'

'16.07.1969'

NULL

# Projection Clause

- Primjeri:

student

| matBr | prez   | postBr |
|-------|--------|--------|
| 100   | Kolar  | 52000  |
| 102   | Horvat | 10000  |
| 105   | Kolar  | 52000  |
| 107   | Ban    | 10000  |

```
SELECT ALL prez
 , postbr
FROM student;
```

≡

```
SELECT prez
 , postbr
FROM student;
```

| prez   | postBr |
|--------|--------|
| Kolar  | 52000  |
| Horvat | 10000  |
| Kolar  | 52000  |
| Ban    | 10000  |

```
SELECT DISTINCT prez
 , postbr
FROM student;
```

| prez   | postBr |
|--------|--------|
| Kolar  | 52000  |
| Horvat | 10000  |
| Ban    | 10000  |

```
SELECT FIRST 2 *
FROM student;
```

| matBr | prez   | postBr |
|-------|--------|--------|
| 102   | Horvat | 10000  |
| 105   | Kolar  | 52000  |

Ne zna se koje dvije n-torce  
će se dobiti kao "prve dvije" -  
poredak n-torki u relaciji (niti u  
SQL tablici) nije definiran

# Izraz (*Expression*)

- Unarni operatori

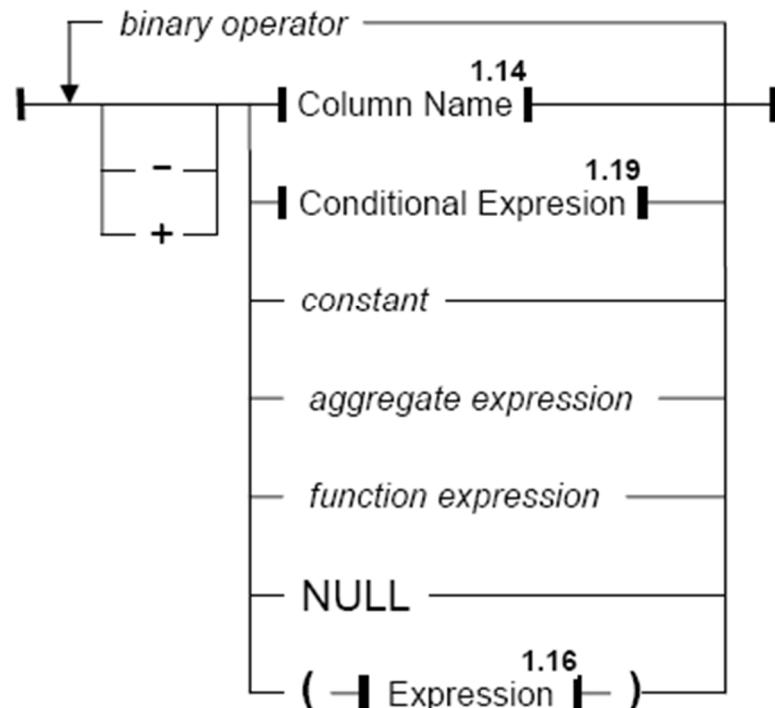
+ -

- Binarni operatori

+ - \* /

|| ulančavanje nizova znakova  
(nadovezivanje, konkatenacija)

## 1.16. Expression



- Redoslijed obavljanja operacija u složenim izrazima određuje se prema istim pravilima kao u programskom jeziku C (implicitni redoslijed obavljanja se može promijeniti upotrebom okruglih zagrada)
- Konverzija tipova podataka tijekom evaluacije izraza obavlja se prema sličnim pravilima kao u programskom jeziku C

# Funkcije (*function expression*)

- **ABS (*num\_expression*)**

- računa absolutnu vrijednost izraza

*num\_expression* – mora biti numerički tip podatka (INTEGER, DECIMAL, FLOAT, ...)

*rezultat funkcije* – tip podatka ovisi o tipu podatka ulaznog argumenta

- **MOD (*dividend, divisor*)**

- računa ostatak cjelobrojnog dijeljenja djeljenika i djelitelja (djelitelj ne smije biti 0)
  - pri računanju uzima se samo cjelobrojni dio argumenata

*dividend (djeljenik)* – numerički tip podatka (INTEGER, DECIMAL, FLOAT, ...)

*divisor (djelitelj)* – numerički tip podatka (INTEGER, DECIMAL, FLOAT, ...)

*rezultat funkcije* – cijeli broj

# Funkcije (*function expression*)

- **ROUND (*expression[, rounding\_factor*)**
  - zaokružuje vrijednost izraza (*expression*)
  - ako se ne navede *rounding\_factor*, uzima se da je njegova vrijednost 0

***expression*** (*izraz koji se zaokružuje*) –

numerički tip podatka (INTEGER, DECIMAL, FLOAT, ...)

***rounding\_factor*** (*preciznost na koju se vrši zaokruživanje*) –

cjelobrojni tip podatka

***rezultat funkcije*** – tip podatka ovisi o tipu podatka ulaznog argumenta (*expression*)

# Funkcije (*function expression*)

- **SUBSTRING (*source\_string FROM start\_position [FOR length]*)**
  - vraća podniz zadanog niza
  - ako se *length* ne navede vraća se podniz koji počinje na *start\_position*, a završava gdje i niz *source\_string*

***source\_string*** – zadani niz čiji se podniz traži funkcijom  
mora biti izraz tipa niza znakova

***start\_position*** – broj koji predstavlja poziciju prvog znaka podniza u zadanim nizu  
*source\_string*;  
mora biti izraz cijelobrojnog tipa

***length(duljina)*** – broj znakova koje funkcija treba vratiti počevši od *start\_position*;  
mora biti izraz cijelobrojnog tipa

# Funkcije (*function expression*)

---

- **UPPER (expression)**

- sva mala slova (a-z) koja se pojavljuju u zadanom nizu *expression* zamjenjuje odgovarajućim velikim slovima (A-Z)

- **LOWER (expression)**

- sva velika slova (A-Z) koja se pojavljuju u zadanom nizu *expression* zamjenjuje odgovarajućim malim slovima (a-z)

***expression*** – zadani niz nad kojim se vrši pretvorba slova  
mora biti izraz tipa niza znakova

# Funkcije (*function expression*)

---

- **TRIM(*source\_expression*)**

- funkcija vraća niz znakova koji nastaje tako da se s početka i kraja niza *source\_expression* izbace sve praznine

***expression*** – zadani niz iz kojeg funkcija izbacuje praznine  
mora biti izraz tipa niza znakova

# Funkcije (*function expression*)

- **CHAR\_LENGTH(*expression*)**
  - funkcija vraća broj znakova u zadanom nizu *expression* uključujući i prateće praznine
- **OCTET\_LENGTH(*expression*)**
  - funkcija vraća broj byte-ova zadanoog niza *expression* uključujući i prateće praznine

***expression*** – mora biti izraz tipa niza znakova

# Funkcije (*function expression*)

---

- **USER**

- funkcija vraća *login* korisnika koji je trenutno prijavljen za rad sa bazom podataka

- **TODAY**

- funkcija vraća današnji datum (dobiven iz operacijskog sustava)

# Funkcije (*function expression*)

- **MDY(*month*, *day*, *year*)**

- funkcija vraća varijablu tipa DATE, odnosno izračunava datum iz tri INTEGER varijable koje predstavljaju dan, mjesec i godinu

|                     |                                                                                                                                        |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b><i>month</i></b> | – broj koji predstavlja broj mjeseca<br>mora biti cijeli broj iz intervala [1,12]                                                      |
| <b><i>day</i></b>   | – broj koji predstavlja redni broj dana u mjesecu<br>mora biti cijeli broj veći od 0 i manji od broja dana u određenom mjesecu         |
| <b><i>year</i></b>  | – broj koji predstavlja godinu<br>mora biti četveroznamenkasti broj cjelobrojnog tipa (ne može se koristiti dvoznamenkasta skraćenica) |

# Funkcije (*function expression*)

- **DAY(*date\_expression*)**
  - funkcija vraća redni broj dana u mjesecu za zadani datum
- **MONTH(*date\_expression*)**
  - funkcija vraća redni broj mjeseca za zadani datum
- **YEAR(*date\_expression*)**
  - funkcija vraća redni broj godine za zadani datum
- **WEEKDAY(*date\_expression*)**
  - funkcija vraća redni broj dana u tjednu za zadani datum  
(0 – nedjelja, 1 – ponedjeljak, 2 – utorak, itd...)

*date\_expression* – izraz tipa DATE

# Uvjet usporedbe (*Comparison Condition*) (2)



| stanjeSklad | sifArt | minS | maxS | stanje |
|-------------|--------|------|------|--------|
| 1           | 10     | 50   | 50   |        |
| 2           | 20     | 60   | 30   |        |
| 3           | 10     | 80   | 5    |        |
| 4           | NULL   | 10   | 15   |        |
| 5           | 10     | 20   | NULL |        |

```
SELECT * FROM stanjeSklad
WHERE stanje BETWEEN minS AND maxS;
```

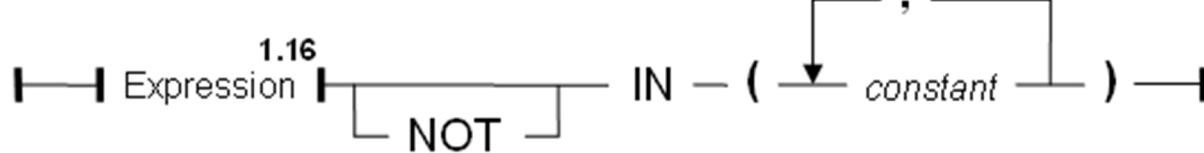
| sifArt | minS | maxS | stanje |
|--------|------|------|--------|
| 1      | 10   | 50   | 50     |
| 2      | 20   | 60   | 30     |

```
SELECT * FROM stanjeSklad
WHERE stanje NOT BETWEEN minS AND maxS;
```

| sifArt | minS | maxS | stanje |
|--------|------|------|--------|
| 3      | 10   | 80   | 5      |
| 4      | NULL | 10   | 15     |

# Uvjet usporedbe (*Comparison Condition*) (3)

## 1.17. IN Condition



student

| matBr | prez   |
|-------|--------|
| 100   | Kolar  |
| 102   | Horvat |
| 103   | Novak  |
| 105   | Horvat |
| 107   | NULL   |
| 109   | Ban    |

```
SELECT * FROM student
WHERE prez IN ('Kolar', 'Horvat');
```

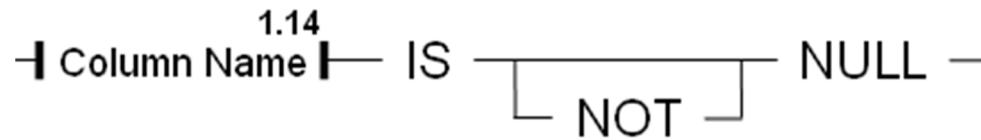
| matBr | prez   |
|-------|--------|
| 100   | Kolar  |
| 102   | Horvat |
| 105   | Horvat |

```
SELECT * FROM student
WHERE prez NOT IN ('Kolar', 'Horvat');
```

| matBr | prez  |
|-------|-------|
| 103   | Novak |
| 109   | Ban   |

- ako *Expression* ima vrijednost NULL, tada je rezultat logička vrijednost *unknown*, bez obzira na vrijednosti navedene u skupu

# Uvjet usporedbe (*Comparison Condition*) (4)



| student | matBr | prez   | postBr |
|---------|-------|--------|--------|
|         | 100   | Kolar  | 52000  |
|         | 102   | Horvat | 10000  |
|         | 105   | Novak  | NULL   |
|         | 107   | Ban    | 10000  |

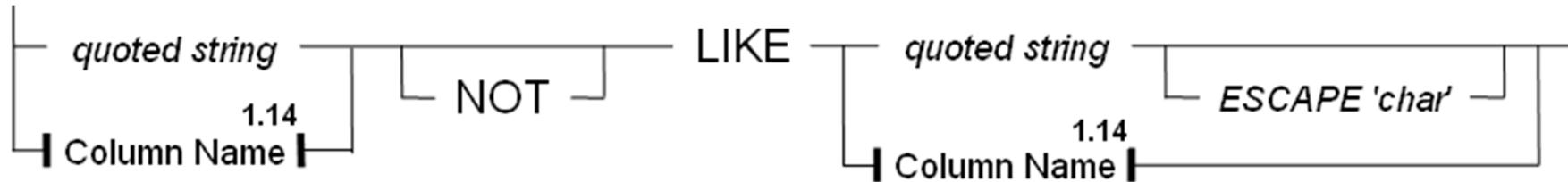
```
SELECT * FROM student
WHERE postBr IS NULL;
```

| matBr | prez  | postBr |
|-------|-------|--------|
| 105   | Novak | NULL   |

```
SELECT * FROM student
WHERE postBr IS NOT NULL;
```

| matBr | prez   | postBr |
|-------|--------|--------|
| 100   | Kolar  | 52000  |
| 102   | Horvat | 10000  |
| 107   | Ban    | 10000  |

# Uvjet usporedbe (*Comparison Condition*) (5)



- služi za ispitivanje zadovoljava li (ili ne zadovoljava) vrijednost atributa ili znakovna konstanta zadani uzorak (*pattern*)
- mogu se koristiti sljedeći *wildcard* znakovi:
  - znak **%** zamjenjuje bilo koju kombinaciju znakova (0 ili više znakova)
  - znak **\_** zamjenjuje točno jedan znak

# Uvjet usporedbe (*Comparison Condition*) (5)

| osoba |         |
|-------|---------|
| matBr | ime     |
| 1     | Matija  |
| 2     | Metka   |
| 3     | Matilda |
| 4     | Ratkec  |
| 5     | Marko   |
| 6     | Ivan    |

```
SELECT * FROM osoba
WHERE ime LIKE 'M%';
```

| matBr | ime     |
|-------|---------|
| 1     | Matija  |
| 2     | Metka   |
| 3     | Matilda |
| 5     | Marko   |

```
SELECT * FROM osoba
WHERE ime LIKE 'Mat%';
```

| matBr | ime     |
|-------|---------|
| 1     | Matija  |
| 3     | Matilda |

```
SELECT * FROM osoba
WHERE ime LIKE 'Ma_k%';
```

| matBr | ime   |
|-------|-------|
| 5     | Marko |

```
SELECT * FROM osoba
WHERE ime LIKE '%tk_';
```

| matBr | ime   |
|-------|-------|
| 2     | Metka |

```
SELECT * FROM osoba
WHERE ime LIKE '%tk%';
```

| matBr | ime    |
|-------|--------|
| 2     | Metka  |
| 4     | Ratkec |

# Uvjet usporedbe (*Comparison Condition*) (5)

tekstovi

| rbr | tekst        |
|-----|--------------|
| 1   | deset %      |
| 2   | pet % kisika |
| 3   | nije pet     |
| 4   | nije_pet     |
| 5   | % i _        |

- znak *char* naveden iza ESCAPE služi za poništavanje specijalnog značenja znakova % ili \_ koji su navedeni neposredno iza znaka *char*

```
SELECT * FROM tekstovi
WHERE tekst LIKE '#%%'
ESCAPE '#';
```

| rbr | tekst |
|-----|-------|
| 5   | % i _ |

```
SELECT * FROM tekstovi
WHERE tekst LIKE '%$%'
ESCAPE '$';
```

| rbr | tekst   |
|-----|---------|
| 1   | deset % |

```
SELECT * FROM tekstovi
WHERE tekst LIKE '%$%%'
ESCAPE '$';
```

| rbr | tekst        |
|-----|--------------|
| 1   | deset %      |
| 2   | pet % kisika |
| 5   | % i _        |

```
SELECT * FROM tekstovi
WHERE tekst LIKE '%!_pet'
ESCAPE '!!';
```

| rbr | tekst    |
|-----|----------|
| 4   | nije_pet |

# Uvjet usporedbe (*Comparison Condition*) (5)

---

- $x \text{ LIKE } 'AB\%'$       *true* za svaki  $x$  koji započinje s AB
  - $x \text{ LIKE } '%AB'$       *true* za svaki  $x$  koji završava s AB
  - $x \text{ LIKE } '%%AB'$       *true* za svaki  $x$  koji završava s AB
- 
- $x \text{ LIKE } 'AB\%CD'$       *true* za svaki  $x$  koji započinje s AB i završava s CD
  - $x \text{ LIKE } '%AB\%'$       *true* za svaki  $x$  koji sadrži AB
  - $x \text{ LIKE } '_AB'$       *true* za svaki  $x$  duljine 3 znaka koji završava s AB
  - $x \text{ LIKE } '__AB'$       *true* za svaki  $x$  duljine 4 znaka koji završava s AB
  - $x \text{ LIKE } 'AB__'$       *true* za svaki  $x$  duljine 4 znaka koji započinje s AB
  - $x \text{ LIKE } '_AB\%'$       *true* za svaki  $x$  koji započinje bilo kojim znakom, nastavlja se sa znakovima AB, te završava s bilo kojim znakovima

# Uvjetni izraz (*Conditional Expression*) (2)

```
SELECT *
 , CASE ocjena
 WHEN 5 THEN 'izvrstan'
 WHEN 4 THEN 'vrlo dobar'
 WHEN 3 THEN 'dobar'
 WHEN 2 THEN 'dovoljan'
 WHEN 1 THEN 'nedovoljan'
 ELSE 'neispravno'
 END AS opis
FROM ispit;
```

| ispit | matBr | ocjena |
|-------|-------|--------|
|       | 100   | 5      |
|       | 102   | 3      |
|       | 103   | 1      |
|       | 107   | NULL   |
|       | 109   | 6      |

| matBr | ocjena | opis       |
|-------|--------|------------|
| 100   | 5      | izvrstan   |
| 102   | 3      | dobar      |
| 103   | 1      | nedovoljan |
| 107   | NULL   | neispravno |
| 109   | 6      | neispravno |

- ako više izraza uz WHEN zadovoljava uvjet jednakosti, rezultat izraza je *Expression* naveden uz prvi WHEN koji zadovoljava uvjet
- ako se ELSE dio izraza ne navede, a niti jedan izraz ne zadovoljava uvjet jednakosti, tada je rezultat izraza NULL vrijednost

## **FROM Clause (2)**

- Zadane su relacije:  $r (\{ A, B \})$     $s (\{ C, D \})$     $t (\{ D, E \})$

$r \times s$

```
SELECT *
 FROM r CROSS JOIN s;
```

$r \triangleright\triangleleft s$   
 $A = C \wedge B \geq D$

```
SELECT *
 FROM r
 INNER JOIN s
 ON A = C AND B >= D;
```

$r \triangleright\triangleleft s$   
 $B = C$

```
SELECT *
 FROM r
 INNER JOIN s
 ON B = C;
```

$\sigma_{D>5}(r \triangleright\triangleleft s)$

```
SELECT *
 FROM r
 INNER JOIN s
 ON B = C
 WHERE D > 5;
```

## **FROM Clause (2)**

- $r (\{ A, B \}) \quad s (\{ C, D \}) \quad t (\{ D, E \}) \quad p (\{ E, F \})$

$(r \times s) \bowtie t$

```
SELECT r.* , s.* , t.E
 FROM r
 CROSS JOIN s
 INNER JOIN t
 ON s.D = t.D;
```

$(s \bowtie t) \bowtie p$

```
SELECT s.* , t.E, p.F
 FROM s
 INNER JOIN t
 ON s.D = t.D
 INNER JOIN p
 ON t.E = p.E;
```

$\sigma_{C=100}(s \bowtie t)$

```
SELECT s.* , t.E
 FROM s
 INNER JOIN t
 ON s.D = t.D
 WHERE C = 100;
```

## **FROM Clause (2)**

- $r (\{ A, B \}) \quad s (\{ C, D \}) \quad t (\{ D, E \}) \quad p (\{ E, F \})$

$(r \times s) * \triangleright \triangleleft t$

```
SELECT r.* , s.* , t.E
FROM r
CROSS JOIN s
LEFT OUTER JOIN t
ON s.D = t.D;
```

$(s * \triangleright \triangleleft * t) \triangleright \triangleleft * p$

```
SELECT s.* , t.D AS D1 , p.*
FROM s
FULL OUTER JOIN t
ON s.D = t.D
RIGHT OUTER JOIN p
ON t.E = p.E;
```

$\sigma_{C=100}(s * \triangleright \triangleleft t)$

```
SELECT s.* , t.E
FROM s
LEFT OUTER JOIN t
ON s.D = t.D
WHERE C = 100;
```

## FROM Clause (2)

- Ako se obavlja operacija spajanja i selekcija, uvjete spajanja treba navesti u ON dijelu, a uvjete selekcije treba navesti u WHERE dijelu SELECT naredbe
  - iako, u slučaju kada se ne koristi vanjsko spajanje, rezultat upita ne ovisi o tome je li uvjet selekcije naveden u ON ili WHERE dijelu naredbe

| student | matBr | prez   | pbrSt |
|---------|-------|--------|-------|
|         | 101   | Kolar  | 10000 |
|         | 102   | Horvat | 21000 |

| mjesto | pbr   | nazMjesto |
|--------|-------|-----------|
|        | 10000 | Zagreb    |
|        | 21000 | Split     |

```
SELECT *
 FROM student
 INNER JOIN mjesto
 ON pbrSt = pbr
 WHERE prez = 'Kolar';
```

```
SELECT *
 FROM student
 INNER JOIN mjesto
 ON pbrSt = pbr
 AND prez = 'Kolar';
```

- u ovom slučaju, oba upita daju isti rezultat

| matBr | prez  | pbrSt | pbr   | nazMjesto |
|-------|-------|-------|-------|-----------|
| 101   | Kolar | 10000 | 10000 | Zagreb    |

## FROM Clause (2)

- Ako se koristi vanjsko spajanje, navođenje uvjeta selekcije u ON dijelu umjesto WHERE dijelu može **bitno** utjecati na rezultat

| student | matBr | prez   | pbrSt |
|---------|-------|--------|-------|
|         | 101   | Kolar  | 10000 |
|         | 102   | Horvat | 21000 |

| mjesto | pbr   | nazMjesto |
|--------|-------|-----------|
|        | 10000 | Zagreb    |
|        | 21000 | Split     |

```
SELECT *
 FROM student
 LEFT OUTER JOIN mjesto
 ON pbrSt = pbr
 WHERE prez = 'Kolar';
```

- Tek nakon obavljenog spajanja prema uvjetu navedenom u ON dijelu naredbe, obavlja se selekcija n-torki prema uvjetu navedenom u WHERE dijelu naredbe

| matBr | prez  | pbrSt | pbr   | nazMjesto |
|-------|-------|-------|-------|-----------|
| 101   | Kolar | 10000 | 10000 | Zagreb    |

## FROM Clause (2)

- Ovdje je prikazan upit sličan prethodnom, ali u kojem je uvjet selekcije napisan na "pogrešnom" mjestu

student

|  | matBr | prez   | pbrSt |
|--|-------|--------|-------|
|  | 101   | Kolar  | 10000 |
|  | 102   | Horvat | 21000 |

mjesto

|  | pbr   | nazMjesto |
|--|-------|-----------|
|  | 10000 | Zagreb    |
|  | 21000 | Split     |

```
SELECT *
 FROM student
LEFT OUTER JOIN mjesto
 ON pbrSt = pbr
 AND prez = 'Kolar';
```

- Ovdje će se pojaviti sve n-torce iz relacije student - uz one n-torce relacije student koje ne zadovoljavaju uvjet spajanja (uočite koji je uvjet spajanja ovdje naveden) dodat će se NULL vrijednosti

| matBr | prez   | pbrSt | pbr   | nazMjesto |
|-------|--------|-------|-------|-----------|
| 101   | Kolar  | 10000 | 10000 | Zagreb    |
| 102   | Horvat | 21000 | NULL  | NULL      |

## **FROM Clause (2)**

---

- **Logički promatrano\***, kada se u upitu spajaju više od dvije relacije, redoslijed spajanja je s lijeva na desno: spajaju se prve dvije relacije, zatim se dobiveni rezultat spaja s trećom navedenom relacijom, zatim se dobiveni rezultat spaja s četvrtom navedenom relacijom, itd.

(\*) konačni rezultat će sigurno odgovarati rezultatu koji bi se dobio kada bi se relacije spajale s lijeva na desno. Fizički promatrano, upit će se možda izvesti drugačijim redoslijedom, ali o tome brine dio SUBP-a koji se naziva optimizator upita

## **FROM Clause (2)**

- ako se ne koristi vanjsko spajanje, redoslijed spajanja je ionako irelevantan, jer vrijedi:

$$( r_1 \bowtie r_2 ) \bowtie r_3 \equiv r_1 \bowtie ( r_2 \bowtie r_3 )$$

- ako se koristi vanjsko spajanje, redoslijed spajanja jest važan jer:

$$( r_1 * \bowtie r_2 ) \bowtie r_3 \neq r_1 * \bowtie ( r_2 \bowtie r_3 )$$

## FROM Clause (2)

| stud |        |       |
|------|--------|-------|
| mbr  | prez   | pbrSt |
| 101  | Horvat | 42000 |
| 102  | Novak  | 21000 |

| mjesto |           |          |
|--------|-----------|----------|
| pbr    | nazMjesto | sifZupMj |
| 42000  | Varaždin  | 7        |
| 21000  | Split     | NULL     |

| zupanija |             |
|----------|-------------|
| sifZup   | nazZup      |
| 7        | Varaždinska |
| 4        | Istarska    |

( stud \*  $\triangleright\triangleleft$  mjesto )  $\triangleright\triangleleft$  zupanija  
pbrSt=pbr                        sifZupMj=sifZup

```
SELECT stud.* , mjesto.* , zupanija.*
FROM stud
LEFT OUTER JOIN mjesto
ON pbrSt = pbr
INNER JOIN zupanija
ON sifZupMj = sifZup;
```

- prvo se spajaju relacije stud i mjesto, a zatim se dobiveni rezultat spaja s relacijom zupanija

| mbr | prez   | pbrSt | pbr   | nazMjesto | sifZupMj | sifZup | nazZup      |
|-----|--------|-------|-------|-----------|----------|--------|-------------|
| 101 | Horvat | 42000 | 42000 | Varaždin  | 7        | 7      | Varaždinska |

## FROM Clause (2)

stud \*▷◁ ( mjesto ▷◁ zupanija )  
pbrSt=pbr                            sifZupMj=sifZup

≡ ( mjesto ▷◁ zupanija ) ▷◁\* stud  
sifZupMj=sifZup                    pbrSt=pbr

da bismo izraz  
relacijske algebре mogli  
napisati u obliku SQL  
naredbe, napisat ћemo  
ga u drugačijem obliku

```
SELECT stud.* , mjesto.* , zupanija.*
FROM mjesto
INNER JOIN zupanija
ON sifZupMj = sifZup
RIGHT OUTER JOIN stud
ON pbrSt = pbr;
```

- prvo se spajaju relacije mjesto i zupanija, a zatim se  
dobiveni rezultat spaja s relacijom stud

| mbr | prez   | pbrSt | pbr   | nazMjesto | sifZupMj | sifZup | nazZup      |
|-----|--------|-------|-------|-----------|----------|--------|-------------|
| 101 | Horvat | 42000 | 42000 | Varaždin  | 7        | 7      | Varaždinska |
| 102 | Novak  | 21000 | NULL  | NULL      | NULL     | NULL   | NULL        |

# Preimenovanje relacija unutar upita

- iako se preimenovanjem relacija može skratiti duljina teksta upita, u praksi se to **ne preporuča** jer upiti postaju manje razumljivi

```
SELECT o.jmbg, prezime, m.pbr, nazMjesto
 FROM osoba AS o
 , mjesto AS m
 , zaposlenje AS z1
 , zupanija AS z2
 WHERE o.jmbg = z1.jmbg
 AND o.pbr = m.pbr
 AND m.sifZup = z2.sifZup
 AND z2.nazZup = 'Varaždinska'
 AND z1.radnoMjesto = 'Dimnjačar'
```

- preimenovanje relacija unutar upita treba se koristiti onda kada se ista relacija pojavljuje u više uloga unutar istog upita

# Paralelno spajanje

- Ispravno rješenje:

| student |       |        |         |
|---------|-------|--------|---------|
| mbr     | prez  | pbrRod | pbrStan |
| 100     | Kolar | 10000  | 21000   |
| 102     | Novak | 21000  | 10000   |
| 103     | Ban   | 10000  | 10000   |

mjesto AS mjestoR

| mjesto |           |
|--------|-----------|
| pbr    | nazMjesto |
| 10000  | Zagreb    |
| 21000  | Split     |

mjesto AS mjestoS

| mjestoR |           |
|---------|-----------|
| pbr     | nazMjesto |
| 10000   | Zagreb    |
| 21000   | Split     |

| mjestoS |           |
|---------|-----------|
| pbr     | nazMjesto |
| 10000   | Zagreb    |
| 21000   | Split     |

```
SELECT mbr, prez
 , pbrRod, mjestoR.nazMjesto AS nazMjestoR
 , pbrStan, mjestoS.nazMjesto AS nazMjestoS
FROM student
 , mjesto AS mjestoR
 , mjesto AS mjestoS
WHERE student.pbrRod = mjestoR.pbr
 AND student.pbrStan = mjestoS.pbr;
```

- u upitu se ista relacija pojavljuje u dvije različite uloge

# Refleksivno spajanje

- Kako dobiti sljedeći rezultat

| sifOrgjed | nazOrgjed  | sifNadorgjed | nazNadorgjed |
|-----------|------------|--------------|--------------|
| 1         | Uprava     | NULL         | NULL         |
| 2         | Odjel A    | 1            | Uprava       |
| 3         | Odjel B    | 1            | Uprava       |
| 4         | Pododjel X | 2            | Odjel A      |
| 5         | Pododjel Y | 2            | Odjel A      |
| 6         | Pododjel Z | 3            | Odjel B      |

- radi se o spajanju relacije same sa sobom
- problem je sličan i slično se rješava kao u slučaju paralelnog spajanja
- relacija orgjed treba se u upitu pojaviti dva puta, jednom u ulozi organizacijske jedinice, a jednom u ulozi njezine nadređene organizacijske jedinice

# Refleksivno spajanje

```
SELECT orgjed.sifOrgjed
 , orgjed.nazOrgjed
 , orgjed.sifNadorgjed
 , nadorgjed.nazOrgjed AS nazNadorgjed
 FROM orgjed
 LEFT OUTER JOIN orgjed AS nadOrgjed
 ON orgjed.sifNadorgjed = nadOrgjed.sifOrgjed;
```

| sifOrgjed | nazOrgjed  | sifNadorgjed | nazNadorgjed |
|-----------|------------|--------------|--------------|
| 1         | Uprava     | NULL         | NULL         |
| 2         | Odjel A    | 1            | Uprava       |
| 3         | Odjel B    | 1            | Uprava       |
| 4         | Pododjel X | 2            | Odjel A      |
| 5         | Pododjel Y | 2            | Odjel A      |
| 6         | Pododjel Z | 3            | Odjel B      |

# Preimenovanje relacija unutar upita

- Još jedan primjer u kojem se koristi preimenovanje relacije
- Ispisati podatke o svim osobama čija je plaća manja od plaće osobe sa šifrom 103

| osoba | sifra | ime  | prez  | placa |
|-------|-------|------|-------|-------|
|       | 100   | Ana  | Novak | 6000  |
|       | 101   | Ana  | Kolar | 5000  |
|       | 102   | Ivan | Kolar | 3000  |
|       | 103   | Ana  | Novak | 5000  |
|       | 104   | Jura | Ban   | 4000  |

```
SELECT osoba.*
 FROM osoba
 INNER JOIN osoba AS osoba103
 ON osoba.placa < osoba103.placa
 AND osoba103.sifra = 103;
```

| sifra | ime  | prez  | placa |
|-------|------|-------|-------|
| 102   | Ivan | Kolar | 3000  |
| 104   | Jura | Ban   | 4000  |

# GROUP BY Clause

ispit

| matBr | nazPredmet    | ocjena |
|-------|---------------|--------|
| 100   | Matematika    | 3      |
| 100   | Programiranje | 2      |
| 100   | Fizika        | 5      |
| 101   | Matematika    | 2      |
| 101   | Programiranje | 2      |
| 101   | Fizika        | 3      |
| 102   | Matematika    | 4      |

```
SELECT nazPredmet AS naziv
 , AVG(ocjena) AS prosjek
 FROM ispit
 GROUP BY nazPredmet;
```

| naziv         | prosjek |
|---------------|---------|
| Matematika    | 3       |
| Programiranje | 2       |
| Fizika        | 4       |

- u GROUP BY nije dopušteno koristiti izraze ili zamjenska imena atributa (*display\_label*)

```
SELECT nazPredmet AS naziv
 , AVG(ocjena)
 FROM ispit
 GROUP BY naziv;
```

# HAVING Clause

| ispit | matBr | nazPredmet    | ocjena |
|-------|-------|---------------|--------|
|       | 100   | Matematika    | 3      |
|       | 100   | Programiranje | 2      |
|       | 100   | Fizika        | 5      |
|       | 101   | Matematika    | 2      |
|       | 101   | Programiranje | 2      |
|       | 101   | Fizika        | 3      |
|       | 102   | Matematika    | 4      |

```
SELECT nazPredmet AS naziv
 , AVG(ocjena) AS prosjek
FROM ispit
GROUP BY nazPredmet;
```

| naziv         | prosjek |
|---------------|---------|
| Matematika    | 3       |
| Programiranje | 2       |
| Fizika        | 4       |

- Kako u rezultatu prikazati samo one grupe koje zadovoljavaju neki uvjet, npr. kako u rezultatu prikazati samo one predmete za koje je prosjek ocjena veći od 2 ?

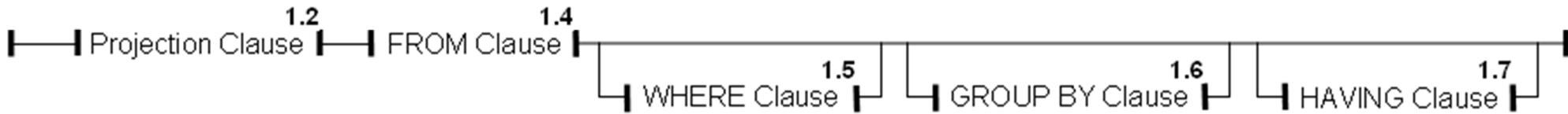
```
SELECT nazPredmet AS naziv
 , AVG(ocjena) AS prosjek
FROM ispit
GROUP BY nazPredmet
HAVING AVG(ocjena) > 2;
```

| naziv      | prosjek |
|------------|---------|
| Matematika | 3       |
| Fizika     | 4       |

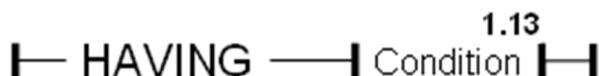
# HAVING Clause

- U *Condition* koji se navodi u HAVING dijelu naredbe dopušteno je u izrazima izvan agregatnih funkcija koristiti samo one atributi koji su navedeni u GROUP BY dijelu naredbe

## 1.1. SELECT Options



## 1.7. HAVING Clause



```
SELECT nazPredmet AS naziv
 , AVG(ocjena) AS prosjek
 FROM ispit
 GROUP BY nazPredmet
 HAVING matBr > 104;
```

# **ORDER BY Clause**

- Koristi se za sortiranje rezultata upita
- Ispisati podatke o položenim ispitima: poredati ih prema ocjenama, tako da se bliže početku liste nalaze studenti s većim ocjenama. Studente koji imaju međusobno jednake ocjene poredati prema prezimenima, tako da se "manja" prezimena ispisuju prije "većih" prezimena (tj. po abecedi)

```
SELECT *
 FROM poloziliProg
 ORDER BY ocjena DESC
 , prez ASC;
```

| poloziliProg |        |        |
|--------------|--------|--------|
| matBr        | prez   | ocjena |
| 100          | Horvat | 3      |
| 107          | Novak  | 3      |
| 102          | Horvat | 5      |
| 101          | Kolar  | 5      |
| 103          | Kolar  | 2      |
| 104          | Horvat | 3      |

| matBr | prez   | ocjena |
|-------|--------|--------|
| 102   | Horvat | 5      |
| 101   | Kolar  | 5      |
| 104   | Horvat | 3      |
| 100   | Horvat | 3      |
| 107   | Novak  | 3      |
| 103   | Kolar  | 2      |

**DESC**

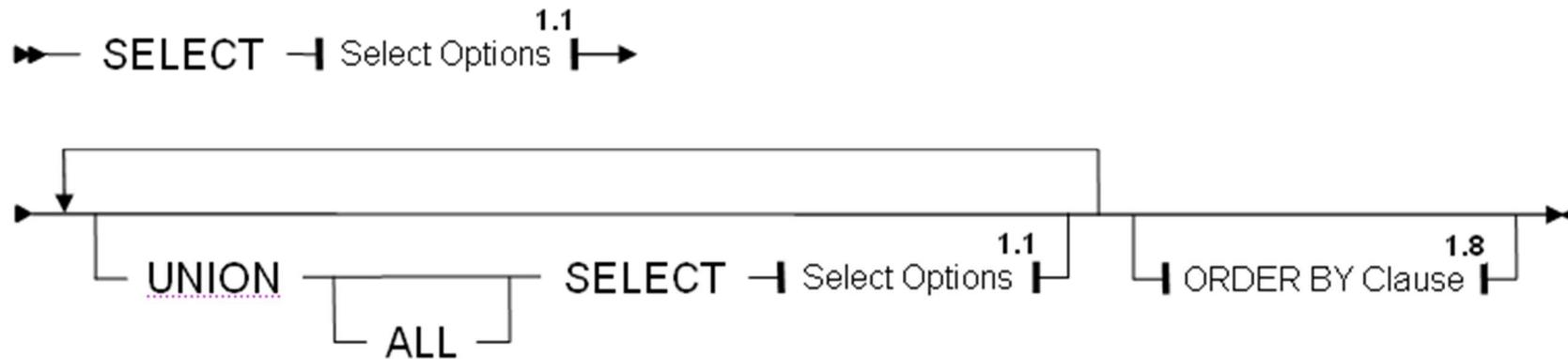
silazno (*descending*)

**ASC**

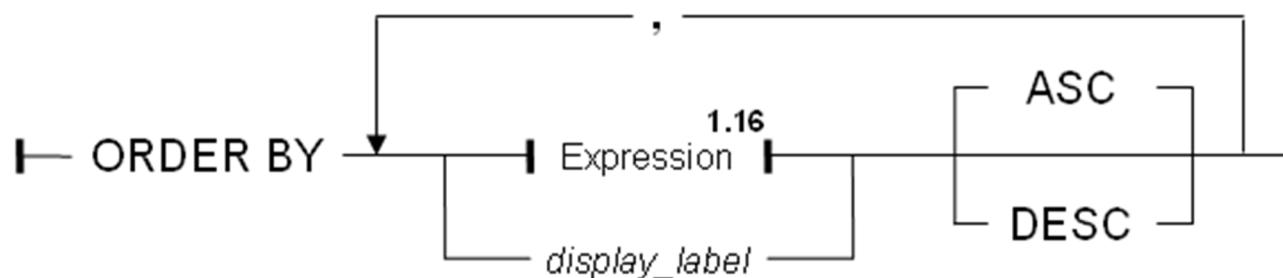
uzlazno (*ascending*)

# ***ORDER BY Clause***

## **1. SELECT Statement**



## **1.8. ORDER BY Clause**



- Ako se smjer sortiranja ne navede, podrazumijeva se uzlazni (ASC) smjer sortiranja

# ***ORDER BY Clause***

---

- U ORDER BY dijelu naredbe mogu se koristiti i izrazi koji nisu navedeni u listi za selekciju
- ORDER BY dio naredbe je jedino mjesto u SELECT naredbi u kojem je dopušteno referencirati se na zamjensko ime atributa (*display\_label*)
- U jednoj SELECT naredbi može se pojaviti samo jedan ORDER BY dio naredbe
  - ako se u SELECT naredbi koristi UNION, ORDER BY se nalazi iza posljednjeg SELECT dijela naredbe
- SQL standard zahtijeva da se NULL vrijednosti pri sortiranju smatraju ili uvijek manjim ili uvijek većim od svih drugih vrijednosti
  - IBM Informix NULL vrijednosti pri sortiranju uvijek tretira kao da su manje od svih ostalih vrijednosti

# ORDER BY Clause

| bodoviMat | mbr | prez   | bodLab | bodMI |
|-----------|-----|--------|--------|-------|
|           | 101 | Novak  | 20     | 30    |
|           | 103 | Horvat | NULL   | 20    |
|           | 107 | Ban    | 10     | 80    |

| bodoviProg | mbr | prez  | bodLab | bodMI |
|------------|-----|-------|--------|-------|
|            | 102 | Kolar | 12     | NULL  |
|            | 104 | Novak | 30     | 0     |

- ispisati podatke o bodovima na lab. vježbama i međuispitu, te ukupnom broju bodova svih studenata, poredati po ukupnom broju bodova: studenti s manjim ukupnim brojem bodova nalaze se bliže početku liste

```
SELECT *, bodLab + bodMI AS ukupno
 FROM bodoviMat
UNION
SELECT *, bodLab + bodMI AS ukupno
 FROM bodoviProg
 ORDER BY ukupno;
```

| mbr | prez   | bodLab | bodMI | ukupno |
|-----|--------|--------|-------|--------|
| 102 | Kolar  | 12     | NULL  | NULL   |
| 103 | Horvat | NULL   | 20    | NULL   |
| 104 | Novak  | 30     | 0     | 30     |
| 101 | Novak  | 20     | 30    | 50     |
| 107 | Ban    | 10     | 80    | 90     |

# "Redoslijed obavljanja" dijelova SELECT naredbe

- |             |             |
|-------------|-------------|
| 1. FROM     | 1. FROM     |
| 2. WHERE    | 2. WHERE    |
| 3. GROUP BY | 3. GROUP BY |
| 4. HAVING   | 4. HAVING   |
| 5. DISTINCT | 5. DISTINCT |

- |             |
|-------------|
| 6. UNION    |
| 7. ORDER BY |



rezultat

- **Logički promatrano**, tj. konačni rezultat će sigurno odgovarati rezultatu koji bi se dobio kada bi se operacije obavljale navedenim redoslijedom. Fizički promatrano, upit će se možda izvesti drugačijim redoslijedom.

# Podupiti (Subqueries)

- Ispravan način:

```
SELECT *
FROM vozilo
WHERE nosivost > (SELECT MAX(tezina)
 FROM teret);
```

vozilo

| sifVoz | nosivost |
|--------|----------|
| 101    | 2500     |
| 102    | 2000     |
| 103    | 800      |
| 104    | 1000     |

podupit

1800

| teret    |        |
|----------|--------|
| sifTeret | tezina |
| 1001     | 1800   |
| 1002     | 1200   |
| 1003     | 1000   |

| sifVoz | nosivost |
|--------|----------|
| 101    | 2500     |
| 102    | 2000     |

$2500 > (\text{SELECT MAX ...}) \rightarrow \text{true}$   
 $2000 > (\text{SELECT MAX ...}) \rightarrow \text{true}$   
 $800 > (\text{SELECT MAX ...}) \rightarrow \text{false}$   
 $1000 > (\text{SELECT MAX ...}) \rightarrow \text{false}$

- U navedenom primjeru je rezultat podupita jednak za svaku n-torku iz relacije vozilo, stoga je (fizički promatrano) rezultat podupita dovoljno izračunati samo jednom tijekom obavljanja upita

# Podupiti (*Subqueries*)

---

- podupit je upit koji je ugrađen u neki drugi upit
  - upit u kojeg je podupit ugrađen naziva se vanjski upit (*outer query*)
  - osim izraza **podupit** (*subquery*), u literaturi se također koristi i izraz **ugniježđeni upit** (*nested query*)
- podupit se u vanjski upit može ugraditi
  - u uvjet (*Condition*) u WHERE dijelu vanjskog upita
  - u uvjet (*Condition*) u HAVING dijelu vanjskog upita
  - u listu za selekciju (*SELECT List*) vanjskog upita
- podupit može sadržavati sve do sada spomenute dijelove SELECT naredbe osim ORDER BY dijela naredbe
- u vanjski upit se može ugraditi više podupita, u svaki od podupita se može ugraditi više podupita, itd.

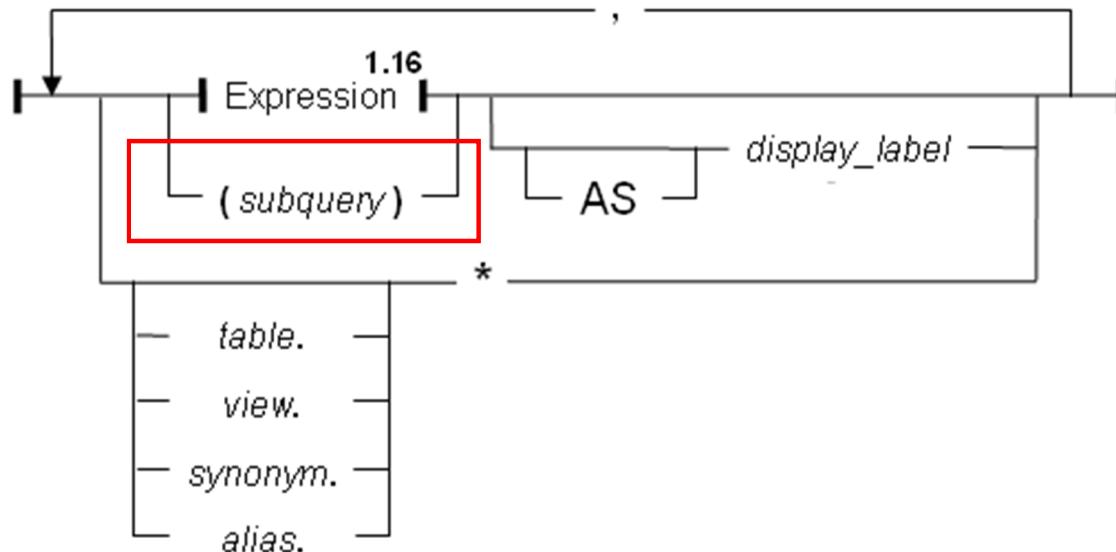
# Skalarni podupit (*Scalar subquery*)

---

- za početak, razmatrat će se najjednostavniji oblik podupita: podupit čiji je rezultat **jedna jednostavna** vrijednost (skalar)
  - npr. podatak tipa: cijeli broj, niz znakova, datum, itd.
- može se reći: rezultat skalarnog podupita je "relacija" stupnja jedan i kardinalnosti jedan
  - vrijednost atributa n-torke dotične "relacije" se u vanjskom upitu koristi kao skalarna vrijednost

# Podupiti u listi za selekciju

## 1.3 SELECT List



| vozilo | sifVoz | nosivost |
|--------|--------|----------|
| 101    | 2500   |          |
| 102    | 2000   |          |
| 103    | 800    |          |
| 104    | 1000   |          |

| teret | sifTeret | tezina |
|-------|----------|--------|
| 1001  | 1800     |        |
| 1002  | 1200     |        |
| 1003  | 1000     |        |

- Ispisati podatke o svim vozilima. Uz svako vozilo ispisati podatak o najvećoj težini tereta

```
SELECT *
 , (SELECT MAX(tezina)
 FROM teret) AS maxTezina
FROM vozilo;
```

| sifVoz | nosivost | maxTezina |
|--------|----------|-----------|
| 101    | 2500     | 1800      |
| 102    | 2000     | 1800      |
| 103    | 800      | 1800      |
| 104    | 1000     | 1800      |

# Podupiti u WHERE dijelu naredbe

- Ispisati podatke o studentima koji stanuju u mjestu Ludbreg

| stud | mbr | prez   | pbrSt |
|------|-----|--------|-------|
|      | 100 | Horvat | 42230 |
|      | 101 | Kolar  | 21000 |
|      | 102 | Novak  | 42230 |

| mjesto | pbr   | nazMjesto |
|--------|-------|-----------|
|        | 42000 | Varaždin  |
|        | 42230 | Ludbreg   |
|        | 21000 | Split     |

```
SELECT * FROM stud
WHERE pbrSt = (SELECT pbr FROM mjesto
 WHERE nazMjesto = 'Ludbreg');
```

42230

- Često se problem može riješiti bez podupita. U konkretnom slučaju, **bolje** rješenje glasi:

```
SELECT stud.*
FROM stud
JOIN mjesto
 ON stud.pbrSt = mjesto.pbr
WHERE nazMjesto = 'Ludbreg';
```

# Podupiti u WHERE dijelu naredbe

- Ispisati podatke o studentima koji su rođeni u mjestu Ludbreg, a stanuju u mjestu Varaždin

| stud | mbr | prez   | pbrRod | pbrSt |
|------|-----|--------|--------|-------|
|      | 100 | Horvat | 42230  | 42230 |
|      | 101 | Kolar  | 21000  | 42000 |
|      | 102 | Novak  | 42230  | 42000 |

| mjesto | pbr   | nazMjesto |
|--------|-------|-----------|
|        | 42000 | Varaždin  |
|        | 42230 | Ludbreg   |
|        | 21000 | Split     |

```
SELECT * FROM stud
```

```
WHERE pbrRod = (SELECT pbr FROM mjesto
 WHERE nazMjesto = 'Ludbreg')
```

```
AND pbrSt = (SELECT pbr FROM mjesto
 WHERE nazMjesto = 'Varaždin');
```

42230

42000

| mbr | prez  | pbrRod | pbrSt |
|-----|-------|--------|-------|
| 102 | Novak | 42230  | 42000 |

- Postoji bolje rješenje u kojem se koriste operacije spajanja i selekcije. Riješiti za vježbu!

# Podupiti u WHERE dijelu naredbe

| stud | mbr    | prez  | pbrSt |
|------|--------|-------|-------|
| 100  | Horvat | 42230 |       |
| 101  | Kolar  | 21000 |       |
| 102  | Novak  | 42230 |       |

| mjesto | pbr      | nazMjesto |
|--------|----------|-----------|
| 42000  | Varaždin |           |
| 42230  | Ludbreg  |           |
| 21000  | Split    |           |

- Ukoliko podupit čiji bi rezultat trebao biti skalar vrati više od jedne n-torce ili više nego jedan atribut, sustav će dojaviti pogrešku

```
SELECT * FROM stud
WHERE pbrSt = (SELECT pbr FROM mjesto
WHERE nazMjesto LIKE '%r%');
```

Pogreška

- Ukoliko podupit čiji bi rezultat trebao biti skalar ne vrati niti jednu n-torku, dobivena skalarna vrijednost će biti NULL vrijednost

```
SELECT * FROM stud
WHERE pbrSt = (SELECT pbr FROM mjesto
WHERE nazMjesto = 'Grad Split');
```

NULL

# Podupiti u HAVING dijelu naredbe

- Primjer:
- Ispisati nazine predmeta za koje se u "D dvoranama" predavanja mogu održati istovremeno za sve grupe. Uz svaki takav predmet ispisati ukupni broj studenata na predmetu

dvorana

| oznDv | kapacitet |
|-------|-----------|
| D1    | 150       |
| D2    | 200       |
| A201  | 80        |

raspored

| predmet     | oznGr | brojSt |
|-------------|-------|--------|
| Matematika  | M1    | 200    |
| Matematika  | M2    | 50     |
| Matematika  | M3    | 50     |
| Fizika      | F1    | 250    |
| Fizika      | F2    | 150    |
| Elektronika | E1    | 50     |
| Elektronika | E2    | 50     |
| Elektronika | E3    | 100    |
| Elektronika | E4    | 150    |

```
SELECT predmet
 , SUM(brojSt) AS ukupnoStud
 FROM raspored
 GROUP BY predmet
 HAVING SUM(brojSt) <=
 (SELECT SUM(kapacitet)
 FROM dvorana
 WHERE oznDv LIKE 'D%');
```

| predmet     | ukupnoStud |
|-------------|------------|
| Matematika  | 300        |
| Elektronika | 350        |

350

# Korelirani podupit (*Correlated subquery*)

```
SELECT oznStr, dopBrSati
 FROM stroj
 WHERE dopBrSati <
 (SELECT SUM(brSatiRada)
 FROM radStroja
 WHERE radStroja.oznStr = stroj.oznStr);
```

| oznStr | dopBrSati |
|--------|-----------|
| S1     | 1000      |
| S3     | 500       |

- upit se (logički promatrano) obavlja na sljedeći način:
  - vanjski upit uzima jednu n-torku iz relacije stroj. Na temelju sadržaja te n-torke i sadržaja relacije radStroja, u podupitu se izračunava suma sati rada dotičnog stroja. Ukoliko je uvjet usporedbe zadovoljen, testirana n-torka se pojavljuje u rezultatu
  - postupak se ponavlja za svaku n-torku relacije stroj

| oznStr | dopBrSati |
|--------|-----------|
| S1     | 1000      |
| S2     | 1500      |
| S3     | 500       |

1000 < (SELECT SUM ... WHERE oznStr = 'S1'  $\Rightarrow$  1100)  $\rightarrow$  true  
1500 < (SELECT SUM ... WHERE oznStr = 'S2'  $\Rightarrow$  1200)  $\rightarrow$  false  
500 < (SELECT SUM ... WHERE oznStr = 'S3'  $\Rightarrow$  600)  $\rightarrow$  true

# Korelirani podupit (*Correlated subquery*)

- Primjer: korelirani podupit u listi za selekciju
- uz svaki stroj koji je korišten više od dopuštenog broja sati, ispisati broj sati korištenja stroja

```
SELECT oznStr
 , dopBrSati
 , (SELECT SUM(brSatiRada)
 FROM radStroja
 WHERE radStroja.oznStr = stroj.oznStr) AS koristenSati
 FROM stroj
 WHERE dopBrSati <
 (SELECT SUM(brSatiRada)
 FROM radStroja
 WHERE radStroja.oznStr = stroj.oznStr);
```

| oznStr | dopBrSati | koristenSati |
|--------|-----------|--------------|
| S1     | 1000      | 1100         |
| S3     | 500       | 600          |

# Korelirani podupit (*Correlated subquery*)

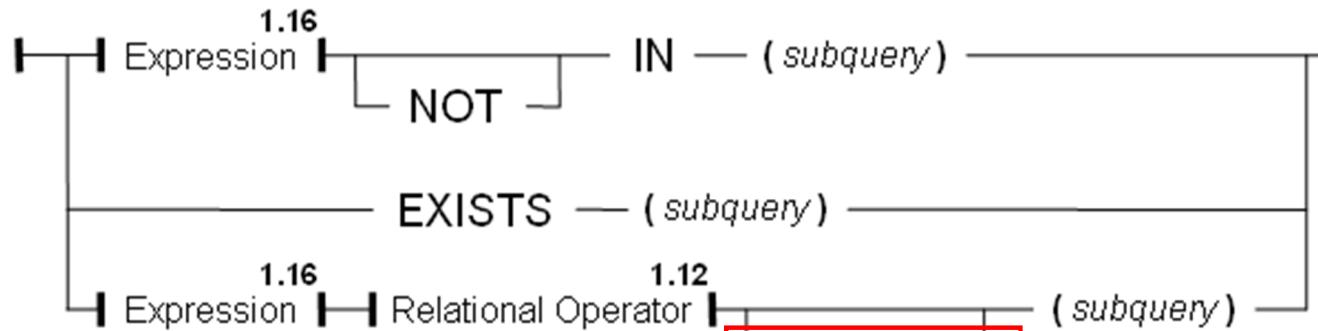
- Rješenje istog problema bez korištenja podupita:

```
SELECT stroj.oznStr
 , dopBrSati
 , SUM(brSatiRada) AS koristenSati
 FROM stroj
 JOIN radStroja
 ON stroj.oznStr = radStroja.oznStr
 GROUP BY stroj.oznStr, dopBrSati
 HAVING dopBrSati < SUM(brSatiRada) ;
```

| oznStr | dopBrSati | koristenSati |
|--------|-----------|--------------|
| S1     | 1000      | 1100         |
| S3     | 500       | 600          |

# Jednostupčani podupit (*Single-column subquery*)

## 1.18. Condition with Subquery



```
SELECT *
 FROM stud
 WHERE prez <> ALL
 (SELECT prez
 FROM nastavnik);
```

|        |
|--------|
| Kolar  |
| Ban    |
| Pernar |

| mbr | ime   | prez   |
|-----|-------|--------|
| 100 | Ivan  | Horvat |
| 102 | Marko | Novak  |

- uvjet selekcije će biti zadovoljen za one n-torke iz relacije stud čija je vrijednost atributa prez različita od vrijednosti svih članova (multi)skupa dobivenog obavljanjem podupita

# Jednostupčani podupit (*Single-column subquery*)

---

- izraz {  $<$  |  $\leq$  |  $=$  |  $\neq$  |  $>$  |  $\geq$  } **ALL** (podupit)
  - *true* ako je izraz {  $<$  |  $\leq$  |  $=$  |  $\neq$  |  $>$  |  $\geq$  } od svih vrijednosti dobivenih podupitom
- izraz {  $<$  |  $\leq$  |  $=$  |  $\neq$  |  $>$  |  $\geq$  } **SOME** (podupit)
  - *true* ako je izraz {  $<$  |  $\leq$  |  $=$  |  $\neq$  |  $>$  |  $\geq$  } od barem jedne vrijednosti dobivene podupitom
- **ANY** je sinonim za **SOME**

# Jednostupčani podupit (*Single-column subquery*)

- Ispisati podatke o dvoranama čiji je kapacitet veći od broja studenata u barem jednoj od grupa

| dvorana | oznDv | kapacitet | grupa | oznGr | brojSt |
|---------|-------|-----------|-------|-------|--------|
|         | D1    | 150       |       | M1    | 100    |
|         | D2    | 120       |       | F1    | 170    |
|         | A201  | 80        |       | E4    | 150    |

```
SELECT *
 FROM dvorana
 WHERE kapacitet > SOME (SELECT brojSt
 FROM grupa);
```

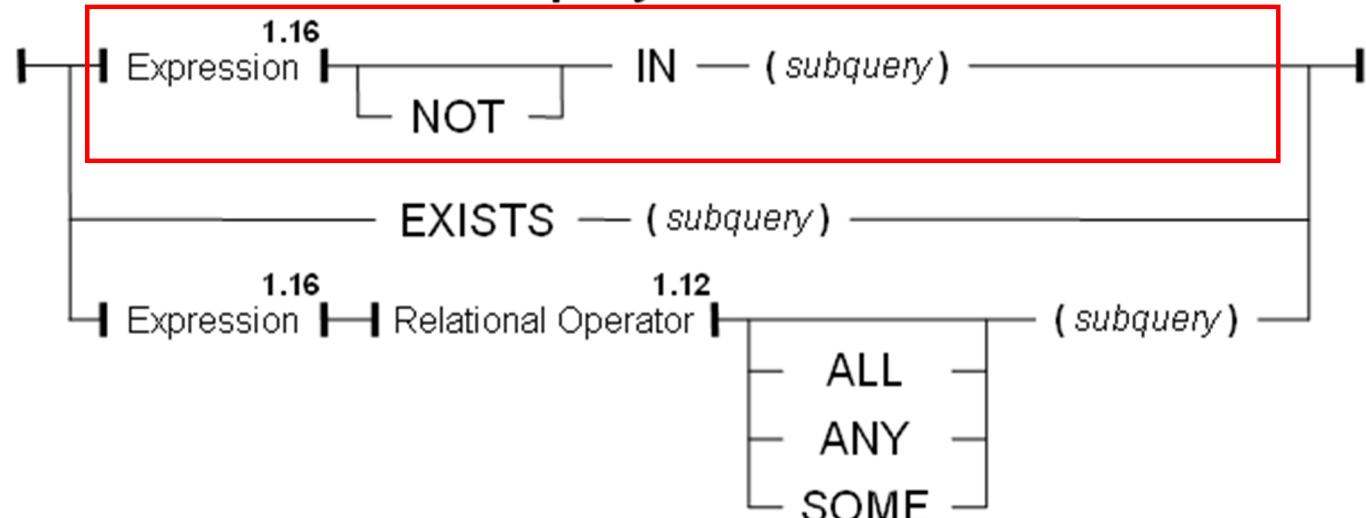
|     |
|-----|
| 100 |
| 170 |
| 150 |

| oznDv | kapacitet |
|-------|-----------|
| D1    | 150       |
| D2    | 120       |

- za vježbu riješiti bez podupita (spajanje, selekcija, projekcija)

# Jednostupčani podupit (*Single-column subquery*)

## 1.18. Condition with Subquery



- izraz **IN** (podupit)
  - *true* ako se u (multi)skupu vrijednosti dobivenih podupitom nalazi barem jedan element jednak vrijednosti izraza
  - ekvivalentno sa: izraz = **SOME** (podupit)
- izraz **NOT IN** (podupit)
  - *true* ako se u (multi)skupu vrijednosti dobivenih podupitom ne nalazi niti jedan element jednak vrijednosti izraza
  - ekvivalentno sa: izraz <> **ALL** (podupit)

# Primjeri: podupiti i NULL vrijednosti

- naročitu pažnju pri korištenju podupita čiji rezultat može sadržavati NULL vrijednosti treba obratiti na uvjete selekcije oblika:

```
WHERE expression relationalOperator ALL (subquery)
```

```
WHERE expression NOT IN (subquery)
```

ukoliko se u rezultatu ovakvih podupita nalazi makar jedna NULL vrijednost, rezultat izračunavanja uvjeta selekcije nikad neće biti *true*

# Operator EXISTS

- Ispisati podatke o studentima koji u akademskoj godini u kojoj su upisali studij nisu položili niti jedan ispit

| stud | mbr | prez   | akGodUpis |
|------|-----|--------|-----------|
|      | 100 | Horvat | 2010      |
|      | 101 | Kolar  | 2010      |
|      | 102 | Novak  | 2011      |
|      | 103 | Ban    | 2010      |

| ispit | mbr | predmet     | akGod | ocjena |
|-------|-----|-------------|-------|--------|
|       | 100 | Matematika  | 2010  | 1      |
|       | 100 | Fizika      | 2011  | 2      |
|       | 100 | Elektronika | 2012  | 4      |
|       | 100 | Matematika  | 2011  | 3      |
|       | 101 | Matematika  | 2010  | 2      |
|       | 101 | Fizika      | 2010  | 5      |
|       | 102 | Matematika  | 2011  | 4      |

```
SELECT *
 FROM stud
 WHERE NOT EXISTS
 (SELECT * FROM ispit
 WHERE ispit.mbr = stud.mbr
 AND akGod = akGodUpis
 AND ocjena > 1);
```

ovdje se npr. moglo napisati samo mbr: dobio bi se jednak rezultat

| mbr | prez   | akGodUpis |
|-----|--------|-----------|
| 100 | Horvat | 2010      |
| 103 | Ban    | 2010      |

# Presjek (u prisustvu NULL vrijednosti)

| polozioMatem |      |       |
|--------------|------|-------|
| mbr          | ime  | prez  |
| 100          | Ivan | Kolar |
| 102          | NULL | Novak |
| 103          | Tea  | Ban   |
| 107          | NULL | NULL  |

| polozioProgr |      |       |
|--------------|------|-------|
| mbr          | ime  | prez  |
| 102          | NULL | Novak |
| 105          | Rudi | Kolar |
| 107          | NULL | NULL  |

**polozioMatem  $\cap$  polozioProgr**

- Paziti: što je kopija n-torce?

| mbr | ime  | prez  |
|-----|------|-------|
| 102 | NULL | Novak |
| 107 | NULL | NULL  |

```
SELECT *
 FROM polozioMatem
 WHERE EXISTS
 (SELECT * FROM polozioProgr
 WHERE
 (polozioProgr.mbr = polozioMatem.mbr OR
 polozioProgr.mbr IS NULL AND polozioMatem.mbr IS NULL)
 AND
 (polozioProgr.ime = polozioMatem.ime OR
 polozioProgr.ime IS NULL AND polozioMatem.ime IS NULL)
 AND
 (polozioProgr.prez = polozioMatem.prez OR
 polozioProgr.prez IS NULL AND polozioMatem.prez IS NULL));
```

# Razlika

polozioMatem

| mbr | ime  | prez   |
|-----|------|--------|
| 100 | Ivan | Kolar  |
| 102 | Ana  | Novak  |
| 103 | Tea  | Ban    |
| 107 | Jura | Horvat |

polozioProgr

| mbr | ime  | prez   |
|-----|------|--------|
| 102 | Ana  | Novak  |
| 105 | Rudi | Kolar  |
| 107 | Jura | Horvat |

studenti koji su položili Matematiku,  
ali nisu položili Programiranje

**polozioMatem \ polozioProgr**

```
SELECT *
 FROM polozioMatem
 WHERE NOT EXISTS
 (SELECT * FROM polozioProgr
 WHERE polozioProgr.mbr = polozioMatem.mbr
 AND polozioProgr.ime = polozioMatem.ime
 AND polozioProgr.prez = polozioMatem.prez) ;
```

- Ispisuju se one n-torce relacije vanjskog upita za koje podupit **ne uspije** pronaći niti jednu n-torku koja ima jednake vrijednosti atributa mbr, ime i prez kao n-torka iz vanjskog upita

| mbr | ime  | prez  |
|-----|------|-------|
| 100 | Ivan | Kolar |
| 103 | Tea  | Ban   |

# INSERT (1), bez navedene liste atributa

- U relaciju se upisuje jedna n-torka pri čemu **vrijednosti svih atributa n-torke moraju biti navedene redoslijedom** kojim su atributi navedeni u CREATE TABLE naredbi kojom je relacija definirana

```
CREATE TABLE stud (
 mbr INTEGER
, ime NCHAR (30)
, prez NCHAR (50)
, stipend CHAR (1) DEFAULT 'N'
, pbrStan INTEGER DEFAULT 10000
);
```

| stud | mbr | ime | prez | stipend | pbrStan |
|------|-----|-----|------|---------|---------|
|      |     |     |      |         |         |

Znači: ako se prilikom unosa nove n-torke ne navede vrijednost za atribut stipend, sustav će kao vrijednost tog atributa postaviti vrijednost N

```
INSERT INTO stud VALUES (
 100
, 'Ivan'
, 'Horvat'
, 'N'
, NULL
);
```

| stud | mbr | ime  | prez   | stipend | pbrStan |
|------|-----|------|--------|---------|---------|
|      | 100 | Ivan | Horvat | N       | NULL    |

# INSERT (1), uz navedenu listu atributa

- U prvom dijelu naredbe navode se imena atributa (i njihov redoslijed) čije će vrijednosti (u odgovarajućem redoslijedu) biti navedene u drugom dijelu naredbe
  - atributi čije vrijednosti nisu navedene u INSERT naredbi, postavljaju se na prepostavljenu (*default*) vrijednost (ukoliko je takva definirana u CREATE TABLE naredbi) ili na NULL vrijednost

```
INSERT INTO stud (
 prez
 , mbr
 , pbrStan
)
VALUES (
 'Kolar'
 , 101
 , 10000
);
```

| stud |      |        |         |         |
|------|------|--------|---------|---------|
| mbr  | ime  | pres   | stipend | pbrStan |
| 100  | Ivan | Horvat | N       | NULL    |

| stud |      |        |         |         |
|------|------|--------|---------|---------|
| mbr  | ime  | pres   | stipend | pbrStan |
| 100  | Ivan | Horvat | N       | NULL    |
| 101  | NULL | Kolar  | N       | 10000   |

# INSERT (2), uz navedenu listu atributa

- u relaciju prosjek upisati podatke o prosječnim ocjenama pojedinih predmeta. Za vrijednost atributa akGodina postaviti NULL vrijednost

| ispit |             |        |
|-------|-------------|--------|
| mbr   | predmet     | ocjena |
| 102   | Elektronika | 1      |
| 102   | Matematika  | 3      |
| 105   | Fizika      | 3      |
| 105   | Matematika  | 4      |
| 107   | Fizika      | 1      |
| 109   | Fizika      | 5      |

```
INSERT INTO prosjek (
 prosOcj
 , predmet
)
SELECT AVG(ocjena)
 , predmet
FROM ispit
GROUP BY predmet;
```

| projek  |         |       |
|---------|---------|-------|
| predmet | prosOcj | akGod |



| projek      |         |       |
|-------------|---------|-------|
| predmet     | prosOcj | akGod |
| Elektronika | 1.0     | NULL  |
| Matematika  | 3.5     | NULL  |
| Fizika      | 3.0     | NULL  |

# DELETE

- iz relacije polozioFiz obrisati n-torce onih studenata koji (sudeći prema podacima iz relacije ispit) nisu položili predmet Fizika

| stud |      |        |       |
|------|------|--------|-------|
| mbr  | ime  | prez   | pbrSt |
| 102  | Ana  | Novak  | 10000 |
| 105  | Rudi | Kolar  | 21000 |
| 107  | Jura | Horvat | 41000 |
| 109  | Tea  | Ban    | 51000 |

| ispit |             |        |
|-------|-------------|--------|
| mbr   | predmet     | ocjena |
| 102   | Elektronika | 1      |
| 102   | Matematika  | 3      |
| 105   | Fizika      | 3      |
| 105   | Matematika  | 4      |
| 107   | Fizika      | 1      |
| 109   | Fizika      | 5      |

| polozioFiz |       |        |
|------------|-------|--------|
| mbr        | imeSt | prezSt |
| 102        | Ana   | Novak  |
| 105        | Rudi  | Kolar  |
| 107        | Jura  | Horvat |
| 109        | Tea   | Ban    |
| 111        | Ivan  | Polak  |

```
DELETE FROM polozioFiz
WHERE mbr NOT IN
(SELECT mbr
FROM ispit
WHERE predmet = 'Fizika'
AND ocjena > 1);
```



| polozioFiz |       |        |
|------------|-------|--------|
| mbr        | imeSt | prezSt |
| 105        | Rudi  | Kolar  |
| 109        | Tea   | Ban    |

# UPDATE

- Bodove na međuispitu uvećati za 10 bodova onim studentima koji time ne bi stekli ukupno (bodLab+bodMI) više od 100 bodova

| bodovi | mbr | prez   | bodLab | bodMI |
|--------|-----|--------|--------|-------|
|        | 101 | Novak  | 30     | 55    |
|        | 102 | Polak  | NULL   | 20    |
|        | 103 | Kolar  | 20     | 10    |
|        | 104 | Ban    | 10     | 80    |
|        | 105 | Horvat | 50     | 49    |
|        | 106 | Seljan | 10     | NULL  |

```
UPDATE bodovi
SET bodMI = bodMI + 10
WHERE bodLab + bodMI + 10 <= 100;
```



| bodovi | mbr | prez   | bodLab | bodMI |
|--------|-----|--------|--------|-------|
|        | 101 | Novak  | 30     | 65    |
|        | 102 | Polak  | NULL   | 20    |
|        | 103 | Kolar  | 20     | 20    |
|        | 104 | Ban    | 10     | 90    |
|        | 105 | Horvat | 50     | 49    |
|        | 106 | Seljan | 10     | NULL  |

# UPDATE

- Bodove na međuispitu uvećati za 2 boda studentima koji time ne bi stekli više od 50 bodova za međuispit, bodove za laboratorij povećati za 3 boda onim studentima koji time ne bi stekli ukupno više od 40 bodova za laboratorij

```
UPDATE bodovi SET
 bodMI = CASE
 WHEN bodMI + 2 <= 50
 THEN bodMI + 2
 ELSE bodMI
 END
, bodLab = CASE
 WHEN bodLab + 3 <= 40
 THEN bodLab + 3
 ELSE bodLab
 END;
```

bodovi

| mbr | prez   | bodLab | bodMI |
|-----|--------|--------|-------|
| 101 | Novak  | 30     | 55    |
| 102 | Polak  | NULL   | 20    |
| 103 | Kolar  | 20     | 10    |
| 104 | Ban    | 10     | 80    |
| 105 | Horvat | 50     | 49    |
| 106 | Seljan | 10     | NULL  |

bodovi

| mbr | prez   | bodLab | bodMI |
|-----|--------|--------|-------|
| 101 | Novak  | 33     | 55    |
| 102 | Polak  | NULL   | 22    |
| 103 | Kolar  | 23     | 12    |
| 104 | Ban    | 13     | 80    |
| 105 | Horvat | 50     | 49    |
| 106 | Seljan | 13     | NULL  |

# UPDATE

- U sintaksnom dijagramu za *SET Clause* može se vidjeti da postoje dva slična, jednako vrijedna oblika:
  - SET atribut1=vrijednost1, atribut2=vrijednost2, ...
  - SET (atribut1, atribut2, ...)=(vrijednost1, vrijednost2, ...)
- Prethodna UPDATE naredba se mogla napisati na sljedeći način:

```
UPDATE bodovi SET
 (bodMI, bodLab) =
 (CASE
 WHEN bodMI + 2 <= 50
 THEN bodMI + 2
 ELSE bodMI
 END
 , CASE
 WHEN bodLab + 3 <= 40
 THEN bodLab + 3
 ELSE bodLab
 END
) ;
```

# UPDATE

- U relaciji mjesto zamijeniti stare poštanske brojeve i nazive (samo onim mjestima za koje postoji opisani novi poštanski brojevi i nazivi)

| mjesto |           | konverzija |         |           |
|--------|-----------|------------|---------|-----------|
| pbr    | nazMjesto | stariPbr   | noviPbr | noviNaziv |
| 41000  | ZAGREB    | 51400      | 52000   | Pazin     |
| 51400  | PAZIN     | 52000      | 52100   | Pula      |
| 52000  | PULA      | 54000      | 31000   | Osijek    |
| 54000  | OSIJEK    |            |         |           |

```
UPDATE mjesto
SET (pbr, nazMjesto) = (
 (SELECT noviPbr
 FROM konverzija
 WHERE konverzija.stariPbr = mjesto.pbr
)
 , (SELECT noviNaziv
 FROM konverzija
 WHERE konverzija.stariPbr = mjesto.pbr
)
)
WHERE pbr IN (SELECT stariPbr
 FROM konverzija);
```

| mjesto |           |
|--------|-----------|
| pbr    | nazMjesto |
| 41000  | ZAGREB    |
| 52000  | Pazin     |
| 52100  | Pula      |
| 31000  | Osijek    |

# UPDATE

- Nastavnicima koji su na ispitima iz BP podijelili (prosječno) najveće ocjene, povećati plaću za 20000

```
UPDATE nast SET placa = placa + 20000
WHERE sifNast IN (
 SELECT sifNast
 FROM ispitBP
 GROUP BY sifNast
 HAVING AVG(ocjena) >= ALL
 (SELECT AVG(ocjena)
 FROM ispitBP
 GROUP BY sifNast)
);
```

| nast | sifNast | prez   | placa |
|------|---------|--------|-------|
|      | 101     | Novak  | 52000 |
|      | 102     | Kolar  | 55000 |
|      | 103     | Horvat | 48000 |
|      | 104     | Ban    | 57000 |

| ispitBP | mbrSt | ocjena | sifNast |
|---------|-------|--------|---------|
|         | 1001  | 5      | 101     |
|         | 1002  | 4      | 101     |
|         | 1003  | 3      | 101     |
|         | 1004  | 2      | 102     |
|         | 1005  | 4      | 102     |
|         | 1006  | 5      | 103     |
|         | 1007  | 5      | 103     |
|         | 1008  | 2      | 103     |
|         | 1009  | 3      | 104     |

| nast | sifNast | prez   | placa |
|------|---------|--------|-------|
|      | 101     | Novak  | 72000 |
|      | 102     | Kolar  | 55000 |
|      | 103     | Horvat | 68000 |
|      | 104     | Ban    | 57000 |

# Oblikovanje sheme baze podataka

---

- cilj: oblikovati shemu baze podataka s dobrim svojstvima
- karakteristike loše koncipirane sheme baze podataka:
  - redundancija (čije su posljedice):
    - neracionalno korištenje prostora za pohranu
    - anomalija unosa
    - anomalija izmjene
    - anomalija brisanja
  - pojava lažnih n-torki

# Anomalija unosa

narudzbaArtikla

| nazProd  | pbr   | nazMjesto | adresa   | brNar | datNar   | sifArtikl | nazArtikl   | kolicina |
|----------|-------|-----------|----------|-------|----------|-----------|-------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2011 | 129       | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2011 | 139       | Albert keks | 2000     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2011 | 129       | Napolitanke | 1200     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2011 | 221       | Domaćica    | 1800     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56 | 4.2.2012 | 129       | Napolitanke | 1100     |

- ne mogu se unijeti podaci o artiklima koje nitko nije naručio
- ne mogu se unijeti podaci o prodavaonicama koje ništa nisu naručile
- ...
- svaki put kad se unosi novi podatak o narudžbi nekog artikla, mora se ponovno upisivati i naziv i mjesto i adresa prodavaonice koja taj artikl naručuje
  - pri tome treba paziti da se podaci za istu prodavaonicu uvijek jednako unesu da bi se zadržala konzistentnost podataka

# Anomalija izmjena

- ako neka prodavaonica promijeni adresu, promjenu adrese potrebno je obaviti na više mjesta da bi se zadržala konzistentnost podataka

| nazProd  | pbr   | nazMjesto | adresa   | brNar | datNar   | sifArtikl | nazArtikl   | kolicina |
|----------|-------|-----------|----------|-------|----------|-----------|-------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2011 | 129       | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2011 | 139       | Albert keks | 2000     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2011 | 129       | Napolitanke | 1200     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2011 | 221       | Domaćica    | 1800     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56 | 4.2.2012 | 129       | Napolitanke | 1100     |

- npr. prodavaonica Konzum-7 se preseli jedan kućni broj dalje od centra

| nazProd  | pbr   | nazMjesto | adresa   | brNar | datNar   | sifArtikl | nazArtikl   | kolicina |
|----------|-------|-----------|----------|-------|----------|-----------|-------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 22 | 13/25 | 1.5.2011 | 129       | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 22 | 13/25 | 1.5.2011 | 139       | Albert keks | 2000     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2011 | 129       | Napolitanke | 1200     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2011 | 221       | Domaćica    | 1800     |
| Konzum-7 | 10000 | Zagreb    | Ilica 22 | 41/56 | 4.2.2012 | 129       | Napolitanke | 1100     |

# Anomalija brisanja

- brisanjem svih narudžbi za neki artikl gube se podaci o artiklu

| nazProd  | pbr   | nazMjesto | adresa   | brNar | datNar   | sifArtikl | nazArtikl   | kolicina |
|----------|-------|-----------|----------|-------|----------|-----------|-------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2011 | 129       | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2011 | 139       | Albert keks | 2000     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2011 | 129       | Napolitanke | 1200     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2011 | 221       | Domaćica    | 1800     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56 | 4.2.2012 | 129       | Napolitanke | 1100     |

- npr. ako se obriše posljednja n-torka o narudžbama artikla Domaćica, podatke o tom artiklu više nećemo imati u bazi podataka

| nazProd  | pbr   | nazMjesto | adresa   | brNar | datNar   | sifArtikl | nazArtikl   | kolicina |
|----------|-------|-----------|----------|-------|----------|-----------|-------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2011 | 129       | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2011 | 139       | Albert keks | 2000     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2011 | 129       | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56 | 4.2.2012 | 129       | Napolitanke | 1100     |

# Kako odrediti zamjenu za loše koncipiranu relacijsku shemu?

---

- proučavanjem značenja podataka (semantike)
- proučavanjem zavisnosti među podacima
- uvođenjem ograničenja koja su ovisna o semantici podataka
- najvažnije su **FUNKCIJSKE ZAVISNOSTI**

# Funkcijske zavisnosti - definicija

---

- Neka je  $r$  relacija sa shemom  $R$  i neka su  $X$  i  $Y$  skupovi atributa,  $X \subseteq R$ ,  $Y \subseteq R$

**Funkcijska zavisnost  $X \rightarrow Y$  vrijedi na shemi  $R$  ukoliko**  
u svim dopuštenim stanjima relacije  $r(R)$  svaki par n-torki  $t_1$  i  $t_2$  koje imaju jednake  $X$ -vrijednosti, također imaju jednake  $Y$ -vrijednosti, odnosno:

$$t_1(X) = t_2(X) \Rightarrow t_1(Y) = t_2(Y)$$

Kratica za funkciju zavisnost je FZ.

# Funkcijske zavisnosti - primjer

- relacija osoba(OSOBA)

| osoba | matBr  | prezime | ime   | postBr | grad |
|-------|--------|---------|-------|--------|------|
| 11234 | Novak  | Josip   | 21000 | Split  |      |
| 12345 | Horvat | Ivan    | 10000 | Zagreb |      |
| 22211 | Kolar  | Ante    | 21000 | Split  |      |
| 33345 | Ban    | Tomo    | 31000 | Osijek |      |
| 23456 | Kolar  | Ana     | 31000 | Osijek |      |

- Funkcijska zavisnost **postBr → grad** vrijedi na shemi OSOBA jer svaki par n-torki koje imaju jednake vrijednosti atributa **postBr** također imaju jednake vrijednosti atributa **grad** (i to vrijedi ne samo za trenutačno stanje relacije, nego za sva dopuštena stanja relacije)
- Vrijedi li funkcija zavisnost **prezime → postBr?**

# Funkcijske zavisnosti

- **Funkcijske zavisnosti proizlaze iz značenja podataka (semantike), a ne iz trenutačnog stanja relacije!**
- Primjer: relacija osoba(OSOBA)

| osoba | matBr | prezime | ime   | pbr |
|-------|-------|---------|-------|-----|
| 11234 | Kolar | Ante    | 21000 |     |
| 22211 | Kolar | Ante    | 31000 |     |
| 33345 | Ban   | Tomo    | 10000 |     |

- promatranjem samo trenutačnog stanja relacije mogli bismo (**pogrešno!**) zaključiti da vrijedi FZ **prezime** → **ime**
- **međutim**, poznavanjem značenja podataka u relaciji možemo zaključiti da je u gore prikazanu relaciju dopušteno unijeti n-torku  $< 76555, \text{Kolar}, \text{Zrinka}, 51000 >$
- ⇒ FZ **prezime** → **ime** **ne vrijedi** na shemi OSOBA

# Armstrongovi aksiomi

## A-1 REFLEKSIVNOST

- **Ako je  $Y \subseteq X$ , tada vrijedi  $X \rightarrow Y$** 
  - uvijek vrijedi  $X \rightarrow X$

PRIMJER:

| osoba(OSOBA) | matBr | prezime | ime   | postBr | grad   |
|--------------|-------|---------|-------|--------|--------|
|              | 11234 | Novak   | Josip | 21000  | Split  |
|              | 12345 | Horvat  | Ivan  | 10000  | Zagreb |
|              | 23456 | Kolar   | Ana   | 31000  | Osijek |
|              | 34567 | Novak   | Josip | 31000  | Osijek |

$$X = \{ \text{prezime, ime} \} \quad Y = \{ \text{prezime} \}$$

$Y \subseteq X \Rightarrow$  u relaciji *osoba* vrijedi i FZ **prezime ime  $\rightarrow$  prezime**

$X \subseteq X \Rightarrow$  u relaciji *osoba* vrijedi i FZ **prezime ime  $\rightarrow$  prezime ime**

# Armstrongovi aksiomi

## A-2 UVEĆANJE

- **Ako u shemi R vrijedi  $X \rightarrow Y$ , tada vrijedi i  $XZ \rightarrow Y$** 
  - možemo uvećati lijevu stranu funkcione zavisnosti

PRIMJER:

**osoba(OSOBA)**

| matBr | prezime | ime   | postBr | grad   |
|-------|---------|-------|--------|--------|
| 11234 | Novak   | Josip | 21000  | Split  |
| 12345 | Horvat  | Ivan  | 10000  | Zagreb |
| 23456 | Kolar   | Ana   | 31000  | Osijek |
| 34567 | Novak   | Josip | 31000  | Osijek |

U relaciji *osoba* vrijedi FZ **matBr  $\rightarrow$  ime**

$\Rightarrow$  u relaciji *osoba* vrijedi i FZ **matBr prezime  $\rightarrow$  ime**

$\Rightarrow$  u relaciji *osoba* vrijedi i FZ **matBr prezime grad  $\rightarrow$  ime**

# Armstrongovi aksiomi

## A-3 TRANZITIVNOST

- **Ako u shemi R vrijedi  $X \rightarrow Y$  i  $Y \rightarrow Z$ , tada vrijedi i  $X \rightarrow Z$** 
  - $X \rightarrow Z$  je tranzitivna zavisnost

PRIMJER:

osoba(OSOBA)

| matBr | prezime | ime   | postBr | grad   |
|-------|---------|-------|--------|--------|
| 11234 | Novak   | Josip | 21000  | Split  |
| 12345 | Horvat  | Ivan  | 10000  | Zagreb |
| 23456 | Kolar   | Ana   | 31000  | Osijek |
| 34567 | Novak   | Josip | 31000  | Osijek |

U relaciji *osoba* vrijede FZ **matBr  $\rightarrow$  postBr** i **postBr  $\rightarrow$  grad**

$\Rightarrow$  u relaciji *osoba* vrijedi i FZ **matBr  $\rightarrow$  grad**

# Pravila koja proizlaze iz Armstrongovih aksioma

## P-1 PRAVILO UNIJE (pravilo o aditivnosti)

- Ako u shemi R vrijedi  $X \rightarrow Y$  i  $X \rightarrow Z$ , tada vrijedi i  $X \rightarrow YZ$

PRIMJER:

| osoba(OSOBA) | matBr | prezime | ime   | postBr | grad   |
|--------------|-------|---------|-------|--------|--------|
|              | 11234 | Novak   | Josip | 21000  | Split  |
|              | 12345 | Horvat  | Ivan  | 10000  | Zagreb |
|              | 23456 | Kolar   | Ana   | 31000  | Osijek |
|              | 34567 | Novak   | Josip | 31000  | Osijek |

U relaciji *osoba* vrijede FZ **matBr → ime** i **matBr → prezime**

⇒ u relaciji *osoba* vrijedi i FZ **matBr → ime prezime**

# Pravila koja proizlaze iz Armstrongovih aksioma

## P-2 PRAVILO DEKOMPOZICIJE (pravilo o projektivnosti)

- **Ako u shemi R vrijedi  $X \rightarrow YZ$ , tada vrijedi i  $X \rightarrow Y$**

PRIMJER:

osoba(OSOBA)

|       | matBr  | prezime | ime   | postBr | grad |
|-------|--------|---------|-------|--------|------|
| 11234 | Novak  | Josip   | 21000 | Split  |      |
| 12345 | Horvat | Ivan    | 10000 | Zagreb |      |
| 23456 | Kolar  | Ana     | 31000 | Osijek |      |
| 34567 | Novak  | Josip   | 31000 | Osijek |      |

U relaciji *osoba* vrijedi FZ **matBr → ime prezime**

⇒ u relaciji *osoba* vrijedi i FZ **matBr → ime**

⇒ u relaciji *osoba* vrijedi i FZ **matBr → prezime**

# Pravila koja proizlaze iz Armstrongovih aksioma

## P-3 PRAVILO PSEUDOTRANZITIVNOSTI

- Ako u shemi R vrijedi  $X \rightarrow Y$  i  $Y \rightarrow Z$ , tada vrijedi i  $X \rightarrow Z$

PRIMJER:

zaposlenje(ZAPOSLENJE)

| jmbg | strSprema | funkcija | zaposlOd | zaposlDo   | placa |
|------|-----------|----------|----------|------------|-------|
| 101  | VSS       | direktor | 1.1.2006 | 31.12.2007 | 10000 |
| 101  | VSS       | tajnik   | 1.1.2008 | 31.12.2008 | 8000  |
| 102  | VŠS       | direktor | 1.1.2009 | 31.12.2009 | 9000  |
| 102  | VŠS       | tajnik   | 1.1.2006 | 31.12.2007 | 7000  |
| 103  | VSS       | direktor | 1.1.2010 | 31.12.2010 | 10000 |
| 101  | VSS       | direktor | 1.1.2011 | 31.12.2011 | 10000 |

U relaciji *zaposlenje* vrijede FZ

**jmbg → strSprema** i **funkcija strSprema → placa**

⇒ u relaciji *zaposlenje* vrijedi i FZ **jmbg funkcija → placa**

# Primjer korištenja aksioma i pravila

---

Uz pretpostavku da na relacijskoj shemi  $R = \{ A, B, C, D, E \}$  vrijedi skup funkcijskih zavisnosti  $F = \{ A \rightarrow BD, B \rightarrow C, D \rightarrow E \}$ , dokazati da vrijedi FZ  $AE \rightarrow AC$ .

Dokaz:

- $A \rightarrow BD$  (P2: dekompozicija)  $\Rightarrow A \rightarrow B$
- $A \rightarrow B \wedge B \rightarrow C$  (A3: tranzitivnost)  $\Rightarrow A \rightarrow C$
- (A1: refleksivnost)  $\Rightarrow A \rightarrow A$
- $A \rightarrow A \wedge A \rightarrow C$  (P1: unija)  $\Rightarrow A \rightarrow AC$
- $A \rightarrow AC$  (A2: uvećanje)  $\Rightarrow AE \rightarrow AC$

# Pravilo o akumulaciji

---

Sljedeće dodatno pravilo omogućuje "algoritamski" pristup rješavanju sličnih zadataka

## PRAVILO O AKUMULACIJI

- **Ako u shemi R vrijedi**
  - $X \rightarrow VZ$  i  $Z \rightarrow W$ , tada vrijedi i  $X \rightarrow VZW$

# Primjer korištenja pravila o akumulaciji

Uz pretpostavku da na relacijskoj shemi  $R = \{ A, B, C, D, E \}$  vrijedi skup funkcijskih zavisnosti  $F = \{ A \rightarrow BD, B \rightarrow C, D \rightarrow E \}$ , dokazati da vrijedi FZ  $AE \rightarrow AC$ .

Označimo lijevu stranu FZ s  $X$  ( $X=AE$ ), a desnu stranu FZ s  $Y$  ( $Y=AC$ ).

Dokaz (primjenom A-1, pravila o akumulaciji i P-2):

1. korak:  $X \rightarrow X$

- (A1: refleksivnost)  $\Rightarrow AE \rightarrow AE$

u sljedećim koracima pomoći pravila akumulacije "uvećavati desnu stranu FZ" sve dok desna strana ne sadrži  $Y$

2. { ▪  $AE \rightarrow AE \wedge A \rightarrow BD$  (akumulacija)  $\Rightarrow AE \rightarrow AE BD$

3. { ▪  $AE \rightarrow AE BD \wedge B \rightarrow C$  (akumulacija)  $\Rightarrow AE \rightarrow AE BDC$

4. { u zadnjem koraku, kad (i ako) desna strana FZ sadrži  $Y$   
▪  $AE \rightarrow AE BDC$  (P2: dekompozicija)  $\Rightarrow AE \rightarrow AC$

# Primjer korištenja pravila o akumulaciji (za vježbu 1)

$$R = \{ L, M, N, P, Q, R \}, \quad F = \{ Q \rightarrow R, M \rightarrow PQ, PQL \rightarrow N \}$$

dokazati da vrijedi FZ  $MLR \rightarrow QN$ .

- (A1: refleksivnost)  $\Rightarrow MLR \rightarrow MLR$
- $MLR \rightarrow MLR \wedge M \rightarrow PQ$  (akumulacija)  $\Rightarrow MLR \rightarrow MLRPQ$
- $MLR \rightarrow MLRPQ \wedge PQL \rightarrow N$  (akumulacija)  $\Rightarrow MLR \rightarrow MLRPQN$
- $MLR \rightarrow MLRPQN$  (P2: dekompozicija)  $\Rightarrow MLR \rightarrow QN$

# Primjer korištenja pravila o akumulaciji (za vježbu 2)

$R = \{ L, M, N, P, Q, R \}$ ,  $F = \{ Q \rightarrow R, M \rightarrow PQ, PQL \rightarrow N \}$   
dokazati da vrijedi FZ  $MQ \rightarrow LN$ .

- (A1: refleksivnost)  $\Rightarrow MQ \rightarrow MQ$
- $MQ \rightarrow MQ \wedge Q \rightarrow R$  (akumulacija)  $\Rightarrow MQ \rightarrow MQR$
- $MQ \rightarrow MQR \wedge M \rightarrow PQ$  (akumulacija)  $\Rightarrow MQ \rightarrow MQRP$
- ne postoji FZ kojom bi se moglo nastaviti "uvećavati desnu stranu"
- $\Rightarrow MQ \rightarrow LN$  ne vrijedi

# Ključ entiteta, ključ relacije

---

- entitet je bilo što, što ima suštinu ili bit i posjeduje značajke s pomoću kojih se može razlučiti od svoje okoline
- ključ entiteta sadrži one atribute koji omogućuju da se pojedini entiteti mogu razlučiti od okoline
- relacijom se opisuje skup entiteta

Ključ relacije je skup atributa koji nedvosmisleno određuje n-torce relacije.

- Ključ relacije ima svojstvo da funkcijски određuje atribute u preostalom dijelu relacije

# Ključ relacijske sheme

- ključ relacijske sheme  $R$  je skup atributa  $K$ ,  $K \subseteq R$ , koji ima sljedeća svojstva:
  1.  $K \rightarrow (R \setminus K)$  (također vrijedi i  $K \rightarrow R$ )
    - ključ funkcijски određuje attribute u preostalom dijelu relacijske sheme
  2. ne postoji  $K' \subset K$  za kojeg vrijedi  $K' \rightarrow R$ 
    - ključ je minimalan skup atributa koji funkcijски određuje attribute u preostalom dijelu relacijske sheme

# Ključ relacije - primjer

| osoba | matBr | prezime | ime   | postBr | grad   |
|-------|-------|---------|-------|--------|--------|
|       | 11234 | Novak   | Josip | 21000  | Split  |
|       | 12345 | Horvat  | Ivan  | 10000  | Zagreb |
|       | 23456 | Kolar   | Ana   | 31000  | Osijek |
|       | 34567 | Novak   | Josip | 10000  | Zagreb |

Ključ:  $K_{\text{OSOBA}} = \{ \text{matBr} \}$

$\text{matBr} \rightarrow \text{prezime}$

$\text{matBr} \rightarrow \text{ime}$

$\text{matBr} \rightarrow \text{postBr}$

$\text{matBr} \rightarrow \text{grad}$

Za  $K = \{ \text{matBr}, \text{prezime} \}$  također vrijedi

$K \rightarrow \{ \text{ime}, \text{postBr}, \text{grad} \},$

ali  $K$  **nije** ključ jer postoji  $K' = \{ \text{matBr} \}$ ,  $K' \subset K$ , za kojeg vrijedi

$K' \rightarrow \{ \text{prezime}, \text{ime}, \text{postBr}, \text{grad} \}$

# Ključevi relacije

- mogući ključevi (*candidate key*)
- primarni ključ (*primary key*) odabire se jedan od mogućih ključeva
- alternativni ključevi (*alternate key*) ostali mogući ključevi

PRIMJER:

| djelatnik | matBr | prezime | ime   | JMBG          |
|-----------|-------|---------|-------|---------------|
|           | 11234 | Novak   | Josip | 1403970330103 |
|           | 12345 | Horvat  | Ivan  | 2812964310267 |
|           | 23456 | Kolar   | Ana   | 0111959335208 |
|           | 34567 | Novak   | Josip | 0301949320319 |

- mogući ključevi:
  - { matBr }
  - { JMBG }
- primarni ključ: { matBr }
- alternativni ključ: { JMBG }

# Struktura relacije

- Relacijska shema sastoji se od:
  - atributa koji su dio ključa (**ključni atributi, ključni dio relacije**)
  - atributa iz zavisnog dijela relacije (**neklučni atributi, neključni dio relacije**)

PRIMJER:

djelatnik

| matBr | prezime | ime   | JMBG          |
|-------|---------|-------|---------------|
| 11234 | Novak   | Josip | 1403970330103 |
| 12345 | Horvat  | Ivan  | 2812964310267 |
| 23456 | Kolar   | Ana   | 0111959335208 |
| 34567 | Novak   | Josip | 0301949320319 |

- primarni ključ: { matBr }
- alternativni ključ: { JMBG }

- ključni atributi, ključni dio relacije:
  - matBr
  - JMBG
- neključni atributi, neključni dio relacije:
  - prezime
  - ime

# Postupci normalizacije

---

- Uočena znanja o međusobnim **funkcijskim zavisnostima** atributa relacije koriste se u postupcima normalizacije.
- **Cilj:**
  - **ukloniti redundanciju**
    - anomalije unosa, izmjene i brisanja
    - neracionalno korištenje prostora za pohranu
  - **spriječiti pojavu lažnih n-torki**
- Postupci normalizacije omogućavaju da se postupno, **točno definiranom metodom**, odredi dobra zamjena za loše koncipiranu relacijsku shemu

**E. F. Codd:**

**“Normalized data base structure: A brief tutorial”**

*Proc. ACM SIGFIDET Workshop on Data Description, Access and Control, 1971*

# Prva normalna forma (1NF)

---

- Definicija:

Relacijska shema je u 1NF ako:

- domene atributa sadrže samo jednostavne (nedjeljive) vrijednosti
- vrijednost svakog atributa je samo jedna vrijednost iz domene tog atributa
- neključni atributi relacije funkcijски ovise o ključu relacije
- Shema baze podataka  $R = \{ R_1, R_2, \dots, R_n \}$  je u 1NF ako je svaka relacijska shema  $R_1, R_2, \dots, R_n$  u 1NF

# Prva normalna forma - primjer

- Poduzeće evidentira podatke o radnicima
  - $\text{RADNIK} = \{ \text{matBr}, \text{prezime}, \text{ime}, \text{datRod}, \text{sifOdjel} \}$

| radnik (RADNIK) | matBr | prezime | ime   | datRod     | sifOdjel |
|-----------------|-------|---------|-------|------------|----------|
|                 | 1111  | Novak   | Ivan  | 28.12.1970 | 50       |
|                 | 1121  | Kolar   | Iva   | 16.10.1965 | 30       |
|                 | 1133  | Horvat  | Krešo | 19.03.1978 | 50       |

$$K_{\text{RADNIK}} = \{ \text{matBr} \}$$

- domene svih atributa sadrže jednostavne (nedjeljive) vrijednosti
- vrijednost svakog atributa je samo jedna vrijednost iz domene tog atributa
- neklučni atributi relacije funkcijski ovise o ključu relacije

⇒ Relacijska shema RADNIK je u 1NF

# Prva normalna forma - primjer

- Poduzeće evidentira podatke o radnicima i njihovoј djeci - korisnicima zdravstvenog osiguranja.

- $\text{RADNIK}_1 = \{ \text{matBr}, \text{prezime}, \text{ime}, \text{imenaDjece} \}$

| $\text{radnik}_1(\text{RADNIK}_1)$         | $\text{matBr}$ | $\text{prezime}$ | $\text{ime}$ | $\text{imenaDjece}$ |
|--------------------------------------------|----------------|------------------|--------------|---------------------|
| $K_{\text{RADNIK}_1} = \{ \text{matBr} \}$ | 1111           | Novak            | Ivan         | Jasna, Vedran       |
|                                            | 1121           | Kolar            | Iva          | Ivan                |
|                                            | 1133           | Horvat           | Krešo        | Petar, Ana, Ivan    |

- domena atributa *imenaDjece* ne sadrži jednostavne (nedjeljive vrijednosti)

⇒ Relacijska shema  $\text{RADNIK}_1$  nije u 1NF

# Prva normalna forma - primjer

- Poduzeće evidentira podatke o radnicima i njihovoј djeci - korisnicima zdravstvenog osiguranja.

- $\text{RADNIK}_2 = \{ \text{matBr}, \text{prezime}, \text{ime}, \text{imeDj}, \text{datRodDj} \}$

| radnik <sub>2</sub> (RADNIK <sub>2</sub> ) | matBr | prezime | ime   | imeDj                | datRodDj                               |
|--------------------------------------------|-------|---------|-------|----------------------|----------------------------------------|
| $K_{\text{RADNIK}_2} = \{ \text{matBr} \}$ | 1111  | Novak   | Ivan  | Jasna<br>Vedran      | 21.01.1995<br>13.12.1997               |
|                                            | 1121  | Kolar   | Iva   | Ivan                 | 23.03.2000                             |
|                                            | 1133  | Horvat  | Krešo | Petar<br>Ana<br>Ivan | 22.02.1998<br>19.09.2000<br>05.11.2002 |

- domene sadrže jednostavne vrijednosti, ali vrijednost atributa *imeDj* nije uvijek samo jedna vrijednost iz domene tog atributa (isto vrijedi i za atribut *datRodDj*)

⇒ Relacijska shema RADNIK<sub>2</sub> nije u 1NF

# Normalizacija na 1NF - izdvajanjem atributa u posebnu relaciju

- u posebnu relaciju izdvaja se skup atributa koji se ponavlja s jednakom kratnošću, zajedno s ključem originalne relacije

$\text{RADNIK}_2 = \{ \text{matBr}, \text{prezime}, \text{ime}, \text{imeDj}, \text{datRodDj} \}$        $K_{\text{RADNIK}_2} = \{ \text{matBr} \}$

| radnik <sub>3</sub> (RADNIK <sub>3</sub> ) | matBr | prezime | ime   |
|--------------------------------------------|-------|---------|-------|
|                                            | 1111  | Novak   | Ivan  |
|                                            | 1121  | Kolar   | Iva   |
|                                            | 1133  | Horvat  | Krešo |

$\text{RADNIK}_3 = \{ \text{matBr}, \text{prezime}, \text{ime} \}$

$K_{\text{RADNIK}_3} = \{ \text{matBr} \}$

| dijete (DIJETE) | matBr | imeDj  | datRodDj   |
|-----------------|-------|--------|------------|
|                 | 1111  | Jasna  | 21.01.1995 |
|                 | 1111  | Vedran | 13.12.1997 |
|                 | 1121  | Ivan.  | 23.03.2000 |
|                 | 1133  | Petar  | 22.02.1998 |
|                 | 1133  | Ana    | 19.09.2000 |
|                 | 1133  | Ivan   | 05.11.2002 |

$\text{DIJETE} = \{ \text{matBr}, \text{imeDj}, \text{datRodDj} \}$

$K_{\text{DIJETE}} = \{ \text{matBr}, \text{imeDj} \}$

- operacija je izvedena bez gubitaka informacija - relacijske sheme imaju zajedničke attribute (matBr), zajednički attribute su ključ u RADNIK<sub>3</sub>

# Normalizacija na 1NF - promjenom ključa

$RADNIK_2 = \{ matBr, prezime, ime, imeDj, datRodDj \}$        $K_{RADNIK_2} = \{ matBr \}$

$RADNIK_4 = \{ matBr, prezime, ime, imeDj, datRodDj \}$

$K_{RADNIK_4} = \{ matBr, imeDj \}$

| radnik <sub>4</sub> (RADNIK <sub>4</sub> ) | matBr | prezime | ime   | imeDj  | datRodDj   |
|--------------------------------------------|-------|---------|-------|--------|------------|
|                                            | 1111  | Novak   | Ivan  | Jasna  | 21.01.1995 |
|                                            | 1111  | Novak   | Ivan  | Vedran | 13.12.1997 |
|                                            | 1121  | Kolar   | Iva   | Ivan   | 23.03.2000 |
|                                            | 1133  | Horvat  | Krešo | Petar  | 22.02.1998 |
|                                            | 1133  | Horvat  | Krešo | Ana    | 19.09.2000 |
|                                            | 1133  | Horvat  | Krešo | Ivan   | 05.11.2002 |

# Druga normalna forma (2NF)

---

- Definicija:

Relacijska shema R je u **2NF** ako je u **1NF** i ako je

- svaki atribut iz zavisnog dijela potpuno funkcijski ovisan o svakom ključu relacije
- Shema baze podataka  $R = \{ R_1, R_2, \dots, R_n \}$  je u **2NF** ako je svaka relacijska shema  $R_1, R_2, \dots, R_n$  u **2NF**

# Potpuna funkcionalna zavisnost

---

- Skup atributa Y **potpuno je funkcijски описан** o skupu atributa X relacijske sheme R ako:
  - Y funkcijski ovisi o X
  - i
  - ne postoji pravi podskup od X koji funkcijski određuje Y

PRIMJER:

Zadan je skup FZ  $F = \{ ABC \rightarrow DE, E \rightarrow F \}$ .

Je li  $\{ D, E \}$  potpuno funkcijski ovisan o  $\{ A, B, C \}$  ?

Da, jer ne postoji skup  $Z \subset \{ A, B, C \}$  takav da  $Z \rightarrow \{ D, E \}$

# Nepotpuna funkcijačka zavisnost

---

Zadana je relacijska shema R i skupovi atributa X i Y iz R, tj.  $X \subseteq R$ ,  $Y \subseteq R$ . Neka u R vrijedi FZ  $X \rightarrow Y$ .

FZ  $X \rightarrow Y$  je **nepotpuna** ako postoji skup atributa Z koji je **pravi podskup od X**, za koji vrijedi  $Z \rightarrow Y$

odnosno

FZ  $X \rightarrow Y$  je **nepotpuna** ako  $(\exists Z) (Z \subset X) : Z \rightarrow Y$

PRIMJER:

Zadan je skup FZ  $F = \{ ABC \rightarrow D, BC \rightarrow E, E \rightarrow D \}$ .

Je li { D } potpuno funkcijски ovisan o { A, B, C } ?

Ne, jer postoji skup { B, C }  $\subset \{ A, B, C \}$  takav da  $\{ B, C \} \rightarrow \{ D \}$

# Druga normalna forma - primjer

$\text{RADNIK}_4 = \{ \text{matBr}, \text{prezime}, \text{ime}, \text{imeDj}, \text{datRodDj} \}$        $K_{\text{RADNIK}_4} = \{ \text{matBr}, \text{imeDj} \}$

| radnik <sub>4</sub> (RADNIK <sub>4</sub> ) | matBr | prezime | ime   | imeDj  | datRodDj   |
|--------------------------------------------|-------|---------|-------|--------|------------|
|                                            | 1111  | Novak   | Ivan  | Jasna  | 21.01.1995 |
|                                            | 1111  | Novak   | Ivan  | Vedran | 13.12.1997 |
|                                            | 1121  | Kolar   | Iva   | Ivan.  | 23.03.2000 |
|                                            | 1133  | Horvat  | Krešo | Petar  | 22.02.1998 |
|                                            | 1133  | Horvat  | Krešo | Ana    | 19.09.2000 |
|                                            | 1133  | Horvat  | Krešo | Ivan   | 05.11.2002 |

- relacijska shema RADNIK<sub>4</sub> zadovoljava 1NF
- **postoji FZ:**  $\text{matBr} \rightarrow \text{prezime } \text{ime}$
- $\text{matBr } \text{imeDj} \rightarrow \text{prezime } \text{ime}$  je nepotpuna FZ!

⇒ Relacijska shema RADNIK<sub>4</sub> nije u 2NF

# Normalizacija na 2NF

- Normalizacijom na 2NF nastaju:
  - relacijska shema koja sadrži skup atributa koji su bili nepotpuno funkcijски ovisni o ključu i dio ključa o kojem su potpuno funkcijски ovisni
  - relacijska shema koja sadrži ključ originalne relacije i skup atributa koji su potpuno funkcijски ovisni o ključu

$RADNIK_5 = \{ \text{matBr}, \text{prezime}, \text{ime} \}$

$K_{RADNIK_5} = \{ \text{matBr} \}$

radnik<sub>5</sub> (RADNIK<sub>5</sub>)

| matBr | prezime | ime   |
|-------|---------|-------|
| 1111  | Novak   | Ivan  |
| 1121  | Kolar   | Iva   |
| 1133  | Horvat  | Krešo |

$DIJETE = \{ \text{matBr}, \text{imeDj}, \text{datRodDj} \}$

$K_{DIJETE} = \{ \text{matBr}, \text{imeDj} \}$

dijete (DIJETE)

| matBr | imeDj  | datRodDj   |
|-------|--------|------------|
| 1111  | Jasna  | 21.01.1995 |
| 1111  | Vedran | 13.12.1997 |
| 1121  | Ivan.  | 23.03.2000 |
| 1133  | Petar  | 22.02.1998 |
| 1133  | Ana    | 19.09.2000 |
| 1133  | Ivan   | 05.11.2002 |

# Normalizacija na 2NF

Neka su  $X, Y, Z, V$  atributi ili skupovi atributa. Zadana je relacijska shema  $R = XYZV$  i na njoj skup funkcijskih zavisnosti  $F = \{ XY \rightarrow ZV, X \rightarrow Z \}$ . Ključ relacije  $K_R = XY$ . R je u 1NF. **Zadovoljava li R 2NF?**

- funkcijска zavisnost  $XY \rightarrow Z$  je nepotpuna  
**R ne zadovoljava 2NF**

Normalizacijom na 2NF shema R se zamjenjuje shemama:

$$R_1 = XZ$$

$$K_{R_1} = X$$

$$R_2 = XYV$$

$$K_{R_2} = XY$$

Relacija  $r(R)$  se normalizacijom na 2NF zamjenjuje projekcijama:

$$r_1 = \pi_{XZ}(r) \quad r_2 = \pi_{XYV}(r)$$

- operacija je izvedena bez gubitaka informacija - relacijske sheme imaju zajedničke attribute (X), zajednički atributi su ključ u  $R_1$ .

# Treća normalna forma (3NF)

---

- Definicija:

Relacijska shema je u **3NF** ako je u **1NF** i ako:

- niti jedan atribut iz zavisnog dijela nije tranzitivno funkcijski ovisan o bilo kojem ključu relacije
- Shema baze podataka  $R = \{ R_1, R_2, \dots, R_n \}$  je u **3NF** ako je svaka relacijska shema  $R_1, R_2, \dots, R_n$  u **3NF**

# Tranzitivna funkcionalna zavisnost

---

Zadano je:

- relacijska shema R,
- skupovi atributa  $X \subseteq R$ ,  $Y \subseteq R$ ,  $Z \subseteq R$

Skup atributa  $Z$  je tranzitivno ovisan o  $X$  ako vrijedi:

- $X \rightarrow Y$ ,  $Y \not\rightarrow X$  i  $Y \rightarrow Z$

# Tranzitivna funkcionalna zavisnost - primjer

Zadana je relacijska shema  $R = \{ A, B, C, D, E, F, G \}$  i skup FZ  
 $F = \{ AB \rightarrow CD, D \rightarrow EF, CD \rightarrow ABG \}$

EF je tranzitivno funkcijски описан о AB, јер

- $AB \rightarrow D, D \not\rightarrow AB, D \rightarrow EF$

G nije tranzitivno funkcijски описан о AB, јер иако

- $AB \rightarrow CD, CD \rightarrow G$
- nije задовољен ујет  $CD \not\rightarrow AB$

# Treća normalna forma - primjer

$\text{OSOBA} = \{ \text{matBr}, \text{prez}, \text{ime}, \text{postBr}, \text{nazMjesto} \}$     $K_{\text{OSOBA}} = \{ \text{matBr} \}$

| osoba (OSOBA) | matBr  | prez  | ime   | postBr | nazMjesto |
|---------------|--------|-------|-------|--------|-----------|
| 1111          | Novak  | Ivan  | 10000 | Zagreb |           |
| 1121          | Kolar  | Iva   | 31000 | Osijek |           |
| 1133          | Horvat | Krešo | 10000 | Zagreb |           |

- Relacijska shema OSOBA zadovoljava 1NF.
  - vrijedi FZ:  $\text{matBr} \rightarrow \text{postBr}$
  - vrijedi FZ:  $\text{postBr} \rightarrow \text{nazMjesto}$
  - ne vrijedi FZ:  $\text{postBr} \rightarrow \text{matBr}$
- $\text{matBr} \rightarrow \text{nazMjesto}$  je tranzitivna zavisnost !
- Relacijska shema OSOBA ne zadovoljava 3NF.

# Normalizacija na 3NF

$\text{OSOBA} = \{ \text{matBr}, \text{prez}, \text{ime}, \text{postBr}, \text{nazMjesto} \}$     $K_{\text{OSOBA}} = \{ \text{matBr} \}$

Normalizacijom na 3NF nastaju:

- relacijska shema koja sadrži skup atributa relacijske sheme OSOBA koji su tranzitivno ovisni o ključu (*nazMjesto*) te srednji skup atributa uočene tranzitivne zavisnosti (*postBr*)
- relacijska shema koja sadrži ključ relacijske sheme OSOBA (*matBr*) i neključne attribute relacijske sheme OSOBA koji nisu tranzitivno ovisni o ključu

$\text{MJESTO} = \{ \text{postBr}, \text{nazMjesto} \}$

$K_{\text{MJESTO}} = \{ \text{postBr} \}$

| mjesto (MJESTO) |           |
|-----------------|-----------|
| postBr          | nazMjesto |
| 10000           | Zagreb    |
| 31000           | Osijek    |

$\text{OSOBA}_1 = \{ \text{matBr}, \text{prezime}, \text{ime}, \text{postBr} \}$

$K_{\text{OSOBA}_1} = \{ \text{matBr} \}$

| osoba <sub>1</sub> (OSOBA <sub>1</sub> ) |         |       |        |
|------------------------------------------|---------|-------|--------|
| matBr                                    | prezime | ime   | postBr |
| 1111                                     | Novak   | Ivan  | 10000  |
| 1121                                     | Kolar   | Iva   | 31000  |
| 1133                                     | Horvat  | Krešo | 10000  |

# Normalizacija na 3NF

Neka su  $X, Y, Z, V$  atributi ili skupovi atributa. Zadana je relacijska shema  $R = XYZV$  i na njoj skup funkcijskih zavisnosti  $F = \{ X \rightarrow YZV, Z \rightarrow V \}$ . Ključ relacije  $K_R = X$ . R je u 1NF. **Zadovoljava li R 3NF?**

- funkcija zavisnost  $X \rightarrow V$  je tranzitivna  
**R ne zadovoljava 3NF**

Normalizacijom na 3NF shema R se zamjenjuje shemama:

$$R_1 = XYZ$$

$$K_{R1} = X$$

$$R_2 = ZV$$

$$K_{R2} = Z$$

Relacija  $r(R)$  se normalizacijom na 3NF zamjenjuje projekcijama:

$$r_1 = \pi_{XYZ}(r) \quad r_2 = \pi_{ZV}(r)$$

- operacija je izvedena bez gubitaka informacija - relacijske sheme imaju zajedničke attribute ( $Z$ ), zajednički attribute su ključ u  $R_2$ .

# Treća normalna forma - komentar

Normalizacija na 2NF nije nužni preuvjet za provođenje normalizacije na 3NF jer se nepotpune FZ mogu promatrati kao tranzitivne FZ.

**Primjer:** zadana je shema  $R = XYZV$  i na njoj skup funkcijskih zavisnosti  $F = \{ XY \rightarrow ZV, X \rightarrow Z \}$ . Ključ relacije  $K_R = XY$ . R je u 1NF, ali nije u 2NF jer postoji nepotpuna FZ  $XY \rightarrow Z$ . Međutim, postoji i tranzitivna funkcionska zavisnost  $XY \rightarrow Z$  ( $XY \rightarrow X \wedge X \rightarrow Z$ ).

Normalizacijom na 3NF shema R se zamjenjuje shemama:

$$R_1 = XZ$$

$$K_{R1} = X$$

$$R_2 = XYV$$

$$K_{R2} = XY$$

$R_1$  i  $R_2$  su u 2NF i 3NF

Preporuka: normalizaciju ipak obavljati postupno  
 $1NF \Rightarrow 2NF \Rightarrow 3NF$

# Normalizacija na 3NF - primjer

$\text{OSOBA}_2 = \{ \text{matBr}, \text{prez}, \text{ime}, \text{JMBG} \}$

| osoba2 (OSOBA2) | matBr | prez   | ime   | JMBG          |
|-----------------|-------|--------|-------|---------------|
|                 | 1111  | Novak  | Ivan  | 1403970330103 |
|                 | 1121  | Kolar  | Iva   | 2812968310267 |
|                 | 1133  | Horvat | Krešo | 0301979320319 |

- postoji FZ:  $\text{matBr} \rightarrow \text{prez} \text{ } \text{ime} \text{ } \text{JMBG}$
- postoje FZ:  $\text{JMBG} \rightarrow \text{prez} \text{ } \text{ime}$  i  $\text{JMBG} \rightarrow \text{matBr}$
- $\text{matBr}$  i  $\text{JMBG}$  su mogući ključevi
- Relacijska shema  $\text{OSOBA}_2$  zadovoljava 3NF.

$$K1_{\text{OSOBA}2} = \{ \text{matBr} \}$$

$$K2_{\text{OSOBA}2} = \{ \text{JMBG} \}$$

# Zadatak 1

- Sadržaj relacije nakon unosa podataka iz prispjelih narudžbi:

narudzbaArtikla

| nazProd  | pbr   | nazMjesto | adresa   | brNar | datNar   | sifArtikl | nazArtikl   | kolicina |
|----------|-------|-----------|----------|-------|----------|-----------|-------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2006 | 129       | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2006 | 139       | Albert keks | 2000     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2006 | 129       | Napolitanke | 1200     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2006 | 221       | Domaćica    | 1800     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56 | 4.2.2007 | 129       | Napolitanke | 1100     |

- Normalizirajte relaciju narudzbaArtikla na 1NF, 2NF, 3NF ako vrijedi da je svaki broj narudžbe jedinstven (ne može se desiti da brojevi narudžbi prispjelih iz različitih prodavaonica budu jednaki)

$\text{brNar} \rightarrow \text{nazProd}$

# Zadatak 1

**NARUDZBAARTIKLA = { nazProd, pbr, nazMjesto, adresa, brNar, datNar, sifArtikl, nazArtikl, kolicina}**

- trenutna vrijednost relacije **narudzbaArtikla (NARUDZBAARTIKLA)** :

narudzbaArtikla

| nazProd  | pbr   | nazMjesto | adresa   | brNar | datNar   | sifArtikl | nazArtikl   | kolicina |
|----------|-------|-----------|----------|-------|----------|-----------|-------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2006 | 129       | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2006 | 139       | Albert keks | 2000     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2006 | 129       | Napolitanke | 1200     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2006 | 221       | Domaćica    | 1800     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56 | 4.2.2007 | 129       | Napolitanke | 1100     |

- odrediti funkcione zavisnosti na temelju značenja podataka
- odrediti primarni ključ relacije (tako da bude zadovoljen uvjet **1NF** prema kojem neključni atributi funkcijски ovise o ključu)
- postupno normalizirati relacijsku shemu NARUDZBAARTIKLA na **2NF** i **3NF**

# Zadatak 1 - 1NF

narudzbaArtikla

| nazProd  | pbr   | nazMjesto | adresa   | brNar | datNar   | sifArtikl | nazArtikl   | kolicina |
|----------|-------|-----------|----------|-------|----------|-----------|-------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2006 | 129       | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25 | 1.5.2006 | 139       | Albert keks | 2000     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2006 | 129       | Napolitanke | 1200     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21 | 7.2.2006 | 221       | Domaćica    | 1800     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56 | 4.2.2007 | 129       | Napolitanke | 1100     |

- Određivanje ključa: bi li brNar bio dobar odabir za ključ?
- Postoje li neključni atributi koji ne ovise o broju narudžbe (brNar)?
- $\text{brNar} \rightarrow \text{nazProd}$   $\text{pbr}$   $\text{nazMjesto}$   $\text{adresa}$   $\text{datNar}$   
međutim:
- $\text{brNar} \not\rightarrow \text{sifArtikl}$        $\text{brNar} \not\rightarrow \text{nazArtikl}$        $\text{brNar} \not\rightarrow \text{kolicina}$
- O kojim atributima funkcijски ovisi atribut  $\text{nazArtikl}$ ?       $\text{sifArtikl} \rightarrow \text{nazArtikl}$
- O kojim atributima funkcijски ovisi atribut  $\text{kolicina}$ ?       $\text{brNar}$   $\text{sifArtikl} \rightarrow \text{kolicina}$   
■ **KLJUČ?**       $\text{sifArtikl} \not\rightarrow \text{kolicina}$

# Zadatak 1 - 1NF

| narudzbaArtikla |       |           |          |              |          |                  |             |          |
|-----------------|-------|-----------|----------|--------------|----------|------------------|-------------|----------|
| nazProd         | pbr   | nazMjesto | adresa   | <u>brNar</u> | datNar   | <u>sifArtikl</u> | nazArtikl   | kolicina |
| Konzum-7        | 10000 | Zagreb    | Ilica 20 | 13/25        | 1.5.2006 | 129              | Napolitanke | 1200     |
| Konzum-7        | 10000 | Zagreb    | Ilica 20 | 13/25        | 1.5.2006 | 139              | Albert keks | 2000     |
| Diona-28        | 21000 | Split     | Bolska 7 | 43-21        | 7.2.2006 | 129              | Napolitanke | 1200     |
| Diona-28        | 21000 | Split     | Bolska 7 | 43-21        | 7.2.2006 | 221              | Domaćica    | 1800     |
| Konzum-7        | 10000 | Zagreb    | Ilica 20 | 41/56        | 4.2.2007 | 129              | Napolitanke | 1100     |

- Prepostavimo  $K = \{ \text{brNar}, \text{sifArtikl} \}$   
Provjerite postoje li neključni atributi koje ključ funkcijski ne određuje.
- $\text{brNar } \text{sifArtikl} \rightarrow \text{nazProd } \text{pbr } \text{nazMjesto } \text{adresa } \text{datNar } \text{nazArtikl } \text{kolicina}$   
postoji li skup  $X \subset \{ \text{brNar}, \text{sifArtikl} \}$  za kojeg vrijedi  $X \rightarrow R$  ?  
 $\Rightarrow \text{NE} \Rightarrow \{ \text{brNar}, \text{sifArtikl} \}$  je mogući ključ

$$K_{\text{NARUDZBAARTIKLA}} = \{ \text{brNar}, \text{sifArtikl} \}$$

- zadovoljen je uvjet 1NF prema kojem neključni atributi funkcijski ovise o ključu

# Zadatak 1 - 2NF

narudzbaArtikla

| nazProd  | pbr   | nazMjesto | adresa   | <u>brNar</u> | datNar   | <u>sifArtikl</u> | nazArtikl   | kolicina |
|----------|-------|-----------|----------|--------------|----------|------------------|-------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25        | 1.5.2006 | 129              | Napolitanke | 1200     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25        | 1.5.2006 | 139              | Albert keks | 2000     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21        | 7.2.2006 | 129              | Napolitanke | 1200     |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21        | 7.2.2006 | 221              | Domaćica    | 1800     |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56        | 4.2.2007 | 129              | Napolitanke | 1100     |

- Postoje li neključni atributi koji ovise o dijelu ključa?

- vrijedi:  $\text{brNar} \rightarrow \text{nazProd}$   $\text{pbr}$   $\text{nazMjesto}$   $\text{adresa}$   $\text{datNar}$   
 $\Rightarrow$  Na koje relacije treba razložiti relaciju narudzbaArtikla?  
Koji su ključevi novonastalih relacija?

$\text{narudzba} = \pi_{\text{nazProd}, \text{pbr}, \text{nazMjesto}, \text{adresa}, \text{brNar}, \text{datNar}} (\text{narudzbaArtikla})$   
 $K_{\text{NARUDZBA}} = \{ \text{brNar} \}$

$\text{stavkaNarudzbe} = \pi_{\text{brNar}, \text{sifArtikl}, \text{nazArtikl}, \text{kolicina}} (\text{narudzbaArtikla})$   
 $K_{\text{STAVKANARUDZBE}} = \{ \text{brNar}, \text{sifArtikl} \}$

# Zadatak 1 - 2NF

brNar → nazProd pbr nazMjesto adresa datNar

| narudzba |       |           |          |              |          |
|----------|-------|-----------|----------|--------------|----------|
| nazProd  | pbr   | nazMjesto | adresa   | <u>brNar</u> | datNar   |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25        | 1.5.2006 |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21        | 7.2.2006 |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56        | 4.2.2007 |

- narudzba ima jednostavan ključ  
⇒ **2NF OK**

| stavkaNarudzbe |                  |             |          |
|----------------|------------------|-------------|----------|
| <u>brNar</u>   | <u>sifArtikl</u> | nazArtikl   | kolicina |
| 13/25          | 129              | Napolitanke | 1200     |
| 13/25          | 139              | Albert keks | 2000     |
| 43-21          | 129              | Napolitanke | 1200     |
| 43-21          | 221              | Domaćica    | 1800     |
| 41/56          | 129              | Napolitanke | 1100     |

**Jesu li relacije narudzba i stavkaNarudzbe u 2NF?**

# Zadatak 1 - 2NF

- Je li stavkaNarudzbe u 2NF?

(postoje li neključni atributi koji ovise o dijelu ključa?)

- Vrijedi:  $sifArtikl \rightarrow nazArtikl$
- ⇒ Na koje relacije treba razložiti relaciju stavkaNarudzbe?  
Koji su ključevi novonastalih relacija?

stavkaNarudzbe

| brNar | sifArtikl | nazArtikl   | kolicina |
|-------|-----------|-------------|----------|
| 13/25 | 129       | Napolitanke | 1200     |
| 13/25 | 139       | Albert keks | 2000     |
| 43-21 | 129       | Napolitanke | 1200     |
| 43-21 | 221       | Domaćica    | 1800     |
| 41/56 | 129       | Napolitanke | 1100     |

$$artikl = \pi_{sifArtikl, nazArtikl}(stavkaNarudzbe)$$

$$K_{ARTIKL} = \{ sifArtikl \}$$

$$stavkaNarudzbe_1 = \pi_{brNar, sifArtikl, kolicina}(stavkaNarudzbe)$$

$$K_{STAVKANARUDZBE1} = \{ brNar, sifArtikl \}$$

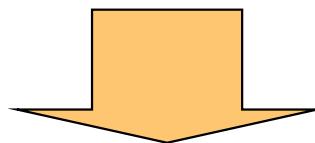
# Zadatak 1 - 2NF

stavkaNarudzbe

| <u>brNar</u> | <u>sifArtikl</u> | nazArtikl   | kolicina |
|--------------|------------------|-------------|----------|
| 13/25        | 129              | Napolitanke | 1200     |
| 13/25        | 139              | Albert keks | 2000     |
| 43-21        | 129              | Napolitanke | 1200     |
| 43-21        | 221              | Domaćica    | 1800     |
| 41/56        | 129              | Napolitanke | 1100     |

sifArtikl → nazArtikl

Jesu li relacije artikl i  
stavkaNarudzbe<sub>1</sub> u 2NF?



artikl

|     | <u>sifArtikl</u> | nazArtikl |
|-----|------------------|-----------|
| 129 | Napolitanke      |           |
| 139 | Albert keks      |           |
| 221 | Domaćica         |           |

2NF O.K.

stavkaNarudzbe<sub>1</sub>

| <u>brNar</u> | <u>sifArtikl</u> | kolicina |
|--------------|------------------|----------|
| 13/25        | 129              | 1200     |
| 13/25        | 139              | 2000     |
| 43-21        | 129              | 1200     |
| 43-21        | 221              | 1800     |
| 41/56        | 129              | 1100     |

2NF O.K.

# Zadatak 1 - 3NF

- Postoje li neključni atributi koji tranzitivno ovise o ključu?

narudzba

| nazProd  | pbr   | nazMjesto | adresa   | <u>brNar</u> | datNar   |
|----------|-------|-----------|----------|--------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25        | 1.5.2006 |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21        | 7.2.2006 |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56        | 4.2.2007 |

Je li  
relacija  
narudzba  
u 3NF?

stavkaNarudzbe<sub>1</sub>

| <u>brNar</u> | <u>sifArtikl</u> | kolicina |
|--------------|------------------|----------|
| 13/25        | 129              | 1200     |
| 13/25        | 139              | 2000     |
| 43-21        | 129              | 1200     |
| 43-21        | 221              | 1800     |
| 41/56        | 129              | 1100     |

3NF O.K.

artikl

| <u>sifArtikl</u> | nazArtikl   |
|------------------|-------------|
| 129              | Napolitanke |
| 139              | Albert keks |
| 221              | Domaćica    |

3NF O.K.

# Zadatak 1 - 3NF

- Postoje li u relaciji narudzba neključni atributi koji tranzitivno ovise o ključu?

| narudzba |       |           |          |              |          |
|----------|-------|-----------|----------|--------------|----------|
| nazProd  | pbr   | nazMjesto | adresa   | <u>brNar</u> | datNar   |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25        | 1.5.2006 |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21        | 7.2.2006 |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56        | 4.2.2007 |

- Vrijedi:  
 $\text{brNar} \rightarrow \text{nazProd}$        $\text{nazProd} \rightarrow \text{pbr}$   $\text{nazProd} \rightarrow \text{nazMjesto}$   $\text{nazProd} \rightarrow \text{adresa}$   
 $\text{nazProd} \not\rightarrow \text{brNar}$

⇒ Na koje relacije treba razložiti relaciju narudzba?

Koji su ključevi novonastalih relacija?

prodavaonica =  $\pi_{\text{nazProd}, \text{pbr}, \text{nazMjesto}, \text{adresa}}(\text{narudzba})$

$K_{\text{PRODAVAONICA}} = \{ \text{nazProd} \}$

$\text{narudzba}_1 = \pi_{\text{brNar}, \text{nazProd}, \text{datNar}}(\text{narudzba})$

$K_{\text{NARUDZBA}_1} = \{ \text{brNar} \}$

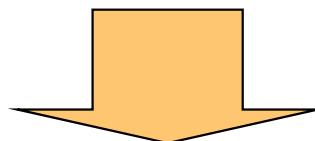
# Zadatak 1 - 3NF

narudzba

| nazProd  | pbr   | nazMjesto | adresa   | <u>brNar</u> | datNar   |
|----------|-------|-----------|----------|--------------|----------|
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 13/25        | 1.5.2006 |
| Diona-28 | 21000 | Split     | Bolska 7 | 43-21        | 7.2.2006 |
| Konzum-7 | 10000 | Zagreb    | Ilica 20 | 41/56        | 4.2.2007 |

$\text{brNar} \rightarrow \text{nazProd}$   
 $\text{nazProd} \not\rightarrow \text{brNar}$

$\text{nazProd} \rightarrow \text{pbr}$   $\text{nazProd} \rightarrow \text{nazMjesto}$   $\text{nazProd} \rightarrow \text{adresa}$



**Jesu li relacije prodavaonica i  
narudzba<sub>1</sub> u 3NF?**

prodavaonica

| <u>nazProd</u> | pbr   | nazMjesto | adresa   |
|----------------|-------|-----------|----------|
| Konzum-7       | 10000 | Zagreb    | Ilica 20 |
| Diona-28       | 21000 | Split     | Bolska 7 |

3NF?

narudzba<sub>1</sub>

| <u>brNar</u> | nazProd  | datNar   |
|--------------|----------|----------|
| 13/25        | Konzum-7 | 1.5.2006 |
| 43-21        | Diona-28 | 7.2.2006 |
| 41/56        | Konzum-7 | 4.2.2007 |

3NF: O.K.

# Zadatak 1 - 3NF

- Postoje li neključni atributi koji tranzitivno ovise o ključu u relaciji prodavaonica?

| prodavaonica   |       |                  |          |
|----------------|-------|------------------|----------|
| <u>nazProd</u> | pbr   | <u>nazMjesto</u> | adresa   |
| Konzum-7       | 10000 | Zagreb           | Ilica 20 |
| Diona-28       | 21000 | Split            | Bolska 7 |

- $\text{nazProd} \rightarrow \text{pbr}$        $\text{pbr} \rightarrow \text{nazMjesto}$
- $\text{pbr} \not\rightarrow \text{nazProd}$

⇒ Na koje relacije treba razložiti relaciju prodavaonica?  
Koji su ključevi novonastalih relacija?

mjesto =  $\pi_{\text{pbr}, \text{nazMjesto}}(\text{prodavaonica})$

$K_{\text{MJESTO}} = \{ \text{pbr} \}$

prodavaonica<sub>1</sub> =  $\pi_{\text{nazProd}, \text{pbr}, \text{adresa}}(\text{prodavaonica})$

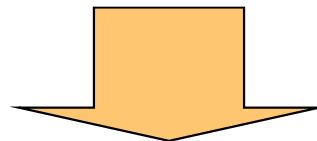
$K_{\text{PRODAVAONICA1}} = \{ \text{nazProd} \}$

# Zadatak 1 - 3NF

prodavaonica

| <u>nazProd</u> | pbr   | nazMjesto | adresa   |
|----------------|-------|-----------|----------|
| Konzum-7       | 10000 | Zagreb    | Ilica 20 |
| Diona-28       | 21000 | Split     | Bolska 7 |

- $\text{nazProd} \rightarrow \text{pbr}$        $\text{pbr} \rightarrow \text{nazMjesto}$
- $\text{pbr} \not\rightarrow \text{nazProd}$



Jesu li relacije mjesto i  
prodavaonica<sub>1</sub> u 3NF?

mjesto

| <u>pbr</u> | nazMjesto |
|------------|-----------|
| 10000      | Zagreb    |
| 21000      | Split     |

3NF: O.K.

prodavaonica<sub>1</sub>

| <u>nazProd</u> | pbr   | adresa   |
|----------------|-------|----------|
| Konzum-7       | 10000 | Ilica 20 |
| Diona-28       | 21000 | Bolska 7 |

3NF: O.K.

# Zadatak 1 - 3NF

mjesto

| pbr   | nazMjesto |
|-------|-----------|
| 10000 | Zagreb    |
| 21000 | Split     |

prodavaonica<sub>1</sub>

| nazProd  | pbr   | adresa   |
|----------|-------|----------|
| Konzum-7 | 10000 | Ilica 20 |
| Diona-28 | 21000 | Bolska 7 |

artikl

| <u>sifArtikl</u> | <u>nazArtikl</u> |
|------------------|------------------|
| 129              | Napolitanke      |
| 139              | Albert keks      |
| 221              | Domaćica         |

narudzba<sub>1</sub>

| <u>brNar</u> | <u>nazProd</u> | <u>datNar</u> |
|--------------|----------------|---------------|
| 13/25        | Konzum-7       | 1.5.2006      |
| 43-21        | Diona-28       | 7.2.2006      |
| 41/56        | Konzum-7       | 4.2.2007      |

stavkaNarudzbe<sub>1</sub>

| <u>brNar</u> | <u>sifArtikl</u> | kolicina |
|--------------|------------------|----------|
| 13/25        | 129              | 1200     |
| 13/25        | 139              | 2000     |
| 43-21        | 129              | 1200     |
| 43-21        | 221              | 1800     |
| 41/56        | 129              | 1100     |

- Shema baze podataka u 3NF sastoji se od relacijskih shema:  
**mjesto, prodavaonica<sub>1</sub>, artikl, narudzba<sub>1</sub>, stavkaNarudzbe<sub>1</sub>**

# Zadatak 5

Zadane su relacijske sheme UREDJAJ i KVAR:

$$\text{UREDJAJ} = \{ \text{mbrUr}, \text{oznVrUr}, \text{nazVrUr}, \text{oznPr}, \text{nazPr} \}$$

$$K_{\text{UREDJAJ}} = \{ \text{mbrUr} \}$$

$$\text{KVAR} = \{ \text{mbrUr}, \text{datKv}, \text{oznVrKv}, \text{opVrKv}, \text{napKv} \}$$

$$K_{\text{KVAR}} = \{ \text{mbrUr}, \text{datKv}, \text{oznVrKv} \}$$

i vrijedi da se za jedan uređaj istog dana može evidentirati više različitih kvarova.

- mbrUr – matični broj uređaja
- oznVrUr – oznaka vrste uređaja
- nazVrUr – naziv vrste uređaja
- oznPr - oznaka proizvođača
- nazPr – naziv proizvođača
- datKv – datum kvara
- oznVrKv – oznaka vrste kvara
- opisVrKv – opis vrste kvara
- napKv – napomena uz kvar (napomena uz konkretni kvar na određenom uređaju određenog datuma)

Relacijske sheme UREDJAJ i KVAR su u 1NF (provjerite!).  
Normalizirati te relacijske sheme na 2NF i 3NF.

## Zadatak 5 – 2NF

2NF?    UREDJAJ = { mbrUr, oznVrUr, nazVrUr, oznPr, nazPr }  
          K<sub>UREDJAJ</sub> = { mbrUr }

UREDJAJ zadovoljava 2NF (zašto?).

KVAR = { mbrUr, datKv, oznVrKv, opVrKv, napKv }  
          K<sub>KVAR</sub> = { mbrUr, datKv, oznVrKv }

Postoje li neključni atributi koji ne ovise o čitavom ključu nego samo o dijelu ključa?  
Normalizirajte relacijsku shemu KVAR na 2NF.

VRSTAKVARA = { oznVrKv, opVrKv }  
          K<sub>VRSTAKVARA</sub> = { oznVrKv }

KVAR<sub>1</sub> = { mbrUr, datKv, oznVrKv, napKv }  
          K<sub>KVAR1</sub> = { mbrUr, datKv, oznVrKv }

# Zadatak 5 – 3NF

Postoje li neključni atributi koji tranzitivno ovise o ključu?

$\text{UREDJAJ} = \{ \text{mbrUr}, \text{oznVrUr}, \text{nazVrUr}, \text{oznPr}, \text{nazPr} \}$

$K_{\text{UREDJAJ}} = \{ \text{mbrUr} \}$



Normalizirajte relacijsku shemu UREDJAJ na 3NF.

$\text{VRSTAUREDJ} = \{ \text{oznVrUr}, \text{nazVrUr} \}$

$K_{\text{VRSTAUREDJ}} = \{ \text{oznVrUr} \}$

$\text{PROIZVODJAC} = \{ \text{oznPr}, \text{nazPr} \}$

$K_{\text{PROIZVODJAC}} = \{ \text{oznPr} \}$

$\text{UREDJAJ}_1 = \{ \text{mbrUr}, \text{oznVrUr}, \text{oznPr} \}$

$K_{\text{UREDJAJ}_1} = \{ \text{mbrUr} \}$

**3NF?**

**OK**

# Zadatak 5 – 3NF

---

$\text{VRSTAKVARA} = \{ \text{oznVrKv}, \text{opVrKv} \}$

3NF OK

$K_{\text{VRSTAKVARA}} = \{ \text{oznVrKv} \}$

$\text{KVAR1} = \{ \text{mbrUr}, \text{datKv}, \text{oznVrKv}, \text{napKv} \}$

3NF OK

$K_{\text{KVAR1}} = \{ \text{mbrUr}, \text{datKv}, \text{oznVrKv} \}$

Shema baze podataka u 3NF sastoji se od relacijskih shema:  
 $\text{VRSTAUREDJ}$ ,  $\text{PROIZVODJAC}$ ,  $\text{UREDJAJ}_1$ ,  $\text{VRSTAKVARA}$ ,  $\text{KVAR}_1$

# Zadatak 6

Zadane su relacijske sheme LINIJA i PROMET:

LINIJA = { lin, sifOdr, nazOdr, vrijPol, trVoz }       $K_{LINIJA} = \{ lin \}$

PROMET = { lin, sifPrij, nazPrij, sifAut, tipAut, datPol, brSjed, brKart }  
 $K_{PROMET} = \{ lin, sifPrij, sifAut, datPol \}$

- lin – broj linije na kojoj se odvija promet
- sifPrij – šifra prijevoznika (poduzeća)
- nazPrij – naziv prijevoznika
- sifAut - šifra autobusa – određuje je prijevoznik
- tipAut – tip autobusa
- brSjed – broj sjedala
- sifOdr – šifra mjesta - odredišta
- nazOdr – naziv mjesta - odredišta
- datPol – datum polaska
- vrijPol – vrijeme polaska
- trVoz – trajanje vožnje
- brKart – broj prodanih karata

Relacijske sheme PROMET i LINIJA su u 1NF (provjeriti!)

# Zadatak 6

---

LINIJA = { lin, sifOdr, nazOdr, vrijPol, trVoz }

$K_{LINIJA} = \{ \text{lin} \}$

PROMET = { lin, sifPrij, nazPrij, sifAut, tipAut, datPol, brSjed, brKart }

$K_{PROMET} = \{ \text{lin}, \text{sifPrij}, \text{sifAut}, \text{datPol} \}$

Normalizirati navedene relacijske sheme na 2NF i 3NF ako vrijedi:

- linija određuje odredište, vrijeme polaska i trajanje vožnje
- istog dana na istoj liniji može prometovati više autobusa (istog ili različitih prijevoznika)
- šifru autobraza određuje prijevoznik – mogu postojati različiti autobusi različitih prijevoznika koji imaju istu šifru
- autobusi istog tipa imaju jednak broj sjedala

## Zadatak 6 – 2NF

2NF? LINIJA = { lin, sifOdr, nazOdr, vrijPol, trVoz } 2NF OK

PROMET = { lin, sifPrij, nazPrij, sifAut, tipAut, datPol, brSjed, brKart }

- šifru autobusa određuje prijevoznik – mogu postojati različiti autobusi različitih prijevoznika koji imaju istu šifru

sifPrij sifAut → tipAut brSjed

Normalizirajte relacijsku shemu PROMET na 2NF.

PRIJEVOZNIK = { sifPrij, nazPrij }

$K_{PRIJEVOZNIK}$  = { sifPrij }

AUTOBUS = { sifPrij, sifAut, tipAut, brSjed }  $K_{AUTOBUS}$  = { sifPrij, sifAut }

PROMET<sub>1</sub> = { lin, sifPrij, sifAut, datPol, brKart }

$K_{PROMET1}$  = { lin, sifPrij, sifAut, datPol }

## Zadatak 6 – 3NF

LINIJA = { lin, sifOdr, nazOdr, vrijPol, trVoz }       $K_{LINIJA} = \{ lin \}$     3NF?

Normalizirajte relacijsku shemu LINIJA na 3NF.

ODREDISTE = { sifOdr, nazOdr }       $K_{ODREDISTE} = \{ sifOdr \}$

$LINIJA_1 = \{ lin, sifOdr, vrijPol, trVoz \}$        $K_{LINIJA1} = \{ lin \}$

PRIJEVOZNIK = { sifPrij, nazPrij }       $K_{PRIJEVOZNIK} = \{ sifPrij \}$     3NF OK

$PROMET_1 = \{ lin, sifPrij, sifAut, datPol, brKart \}$

$K_{PROMET1} = \{ lin, sifPrij, sifAut, datPol \}$

3NF?

OK

## Zadatak 6 – 3NF

AUTOBUS = { sifPrij, sifAut, tipAut, brSjed }

3NF?

$K_{\text{AUTOBUS}} = \{ \text{sifPrij}, \text{sifAut} \}$

- autobusi istog tipa imaju jednak broj sjedala

Normalizirajte relacijsku shemu AUTOBUS na 3NF.

TIPAUTOB = { tipAut, brSjed }

$K_{\text{TIPAUTOB}} = \{ \text{tipAut} \}$

$\text{AUTOBUS}_1 = \{ \text{sifPrij}, \text{sifAut}, \text{tipAut} \}$

$K_{\text{AUTOBUS}_1} = \{ \text{sifPrij}, \text{sifAut} \}$

3NF OK

Shema baze podataka u 3NF sastoji se od relacijskih shema:

ODREDISTE, LINIJA<sub>1</sub>, PRIJEVOZNIK,  
PROMET<sub>1</sub>, TIPAUTOB, AUTOBUS<sub>1</sub>

# Neporedana (*heap*) datoteka

- zapis se upisuje na bilo koje slobodno mjesto u datoteci
- pristup podacima (dohvat podatka sa zadanom vrijednošću ključa pretrage) moguć je isključivo linearnim pretraživanjem



- koristi se za relacije s malim brojem n-torki ili u relacijama čiji se podaci uvijek obrađuju slijedno

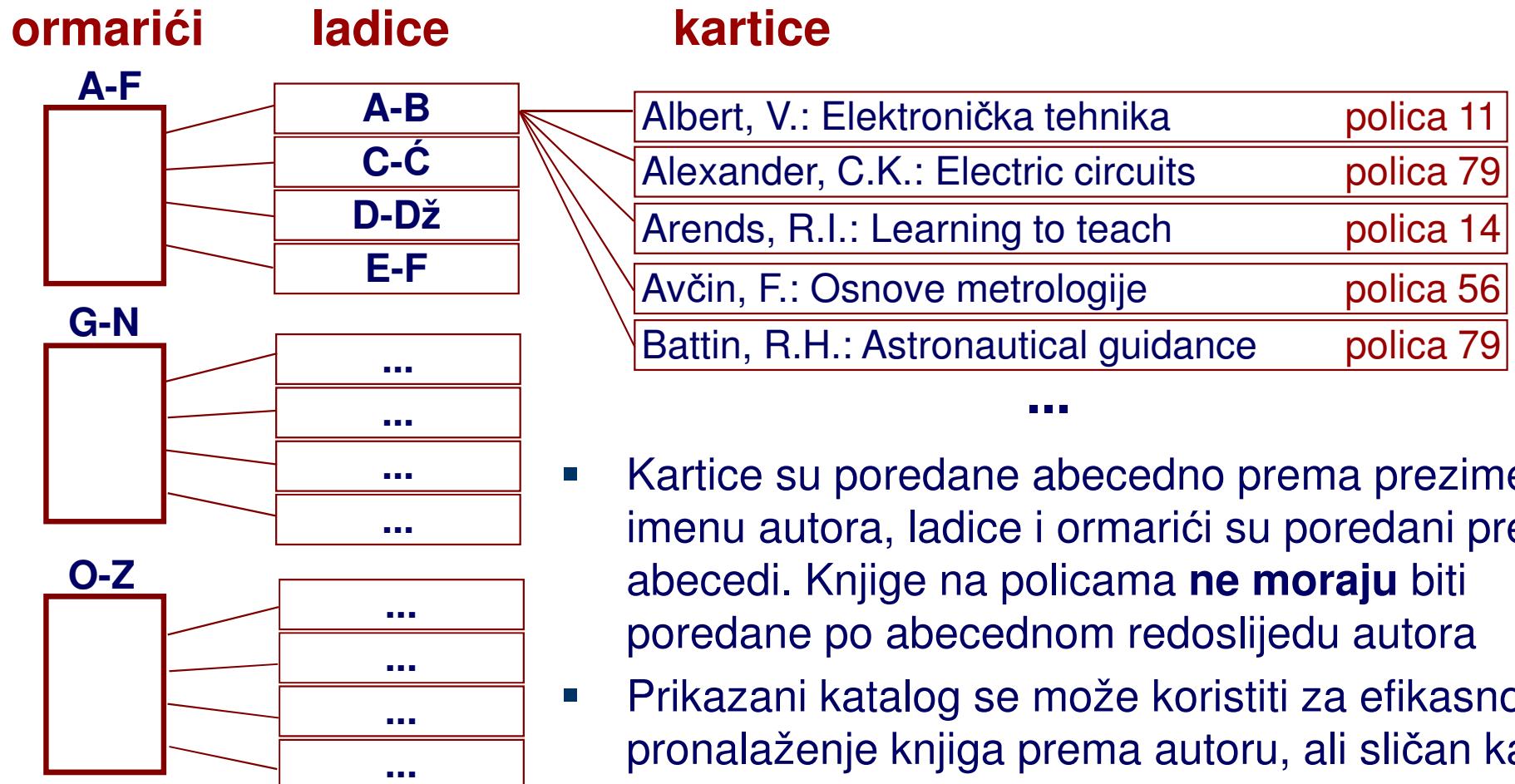
# Neporedana (*heap*) datoteka

---

- Dohvat zapisa prema ključu pretrage
  - prema primarnom ili alternativnom ključu
    - u prosjeku je potrebno obaviti  $n/2$  U/I operacija (n predstavlja broj fizičkih blokova u kojima su pohranjeni logički zapisi odnosno n-torce)
    - još gore: u slučaju kada traženi zapis ne postoji, sustav će morati obaviti n U/I operacija
  - prema ostalim ključevima pretrage ili prema zadanim granicama intervala
    - potrebno je obaviti prijenos n fizičkih blokova

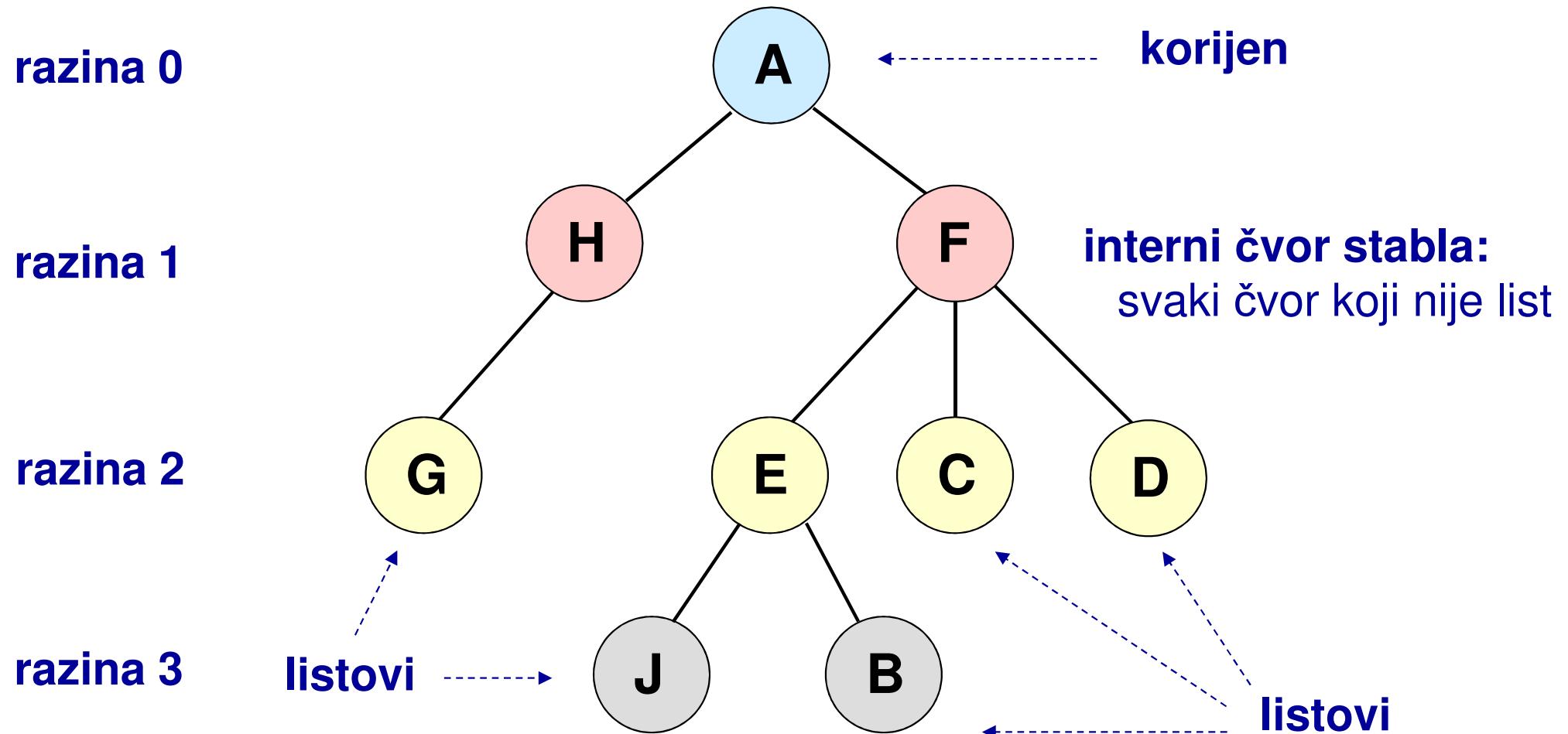
# B-stabla

- Ideja se temelji na izgradnji indeksnog kazala na više razina: slično kao kod kataloga u papirnatom obliku u knjižnicima (danас se rijetko koriste)



- Kartice su poredane abecedno prema prezimenu i imenu autora, ladice i ormarići su poredani prema abecedi. Knjige na policama **ne moraju** biti poredane po abecednom redoslijedu autora
- Prikazani katalog se može koristiti za efikasno pronalaženje knjiga prema autoru, ali sličan katalog se može izgraditi i za brzo pronalaženje knjiga prema naslovu ili prema nekim drugim pojmovima.

# Stablo kao struktura podataka

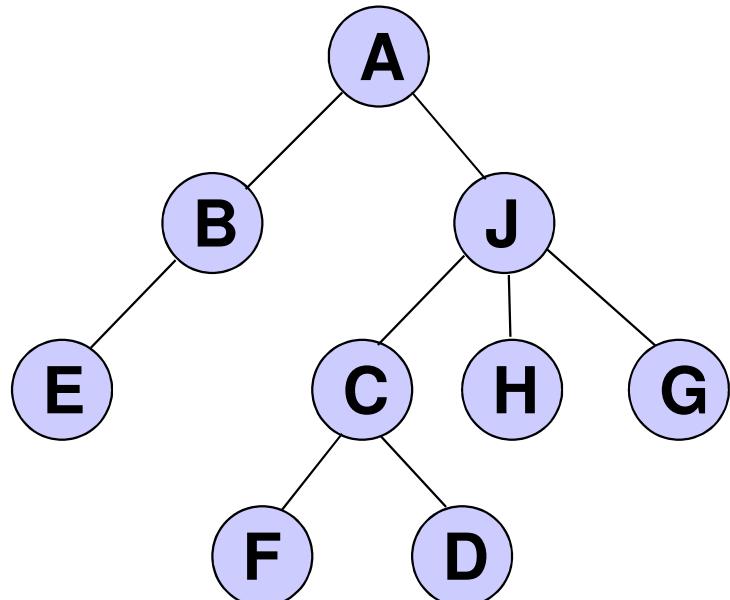


- **razina čvora (level):** duljina puta od korijena do čvora
- **dubina stabla (depth):** najveća duljina puta od korijena do lista
- **red stabla (order):** najveći broj djece koje čvor može imati

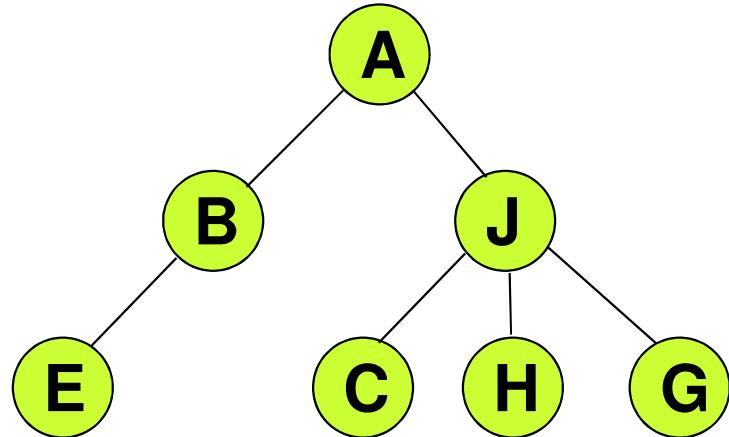
# Stablo kao struktura podataka

Stablo je **balansirano (*balanced*)** ukoliko je duljina puta od korijena do lista jednaka za svaki list u stablu

stablo nije balansirano

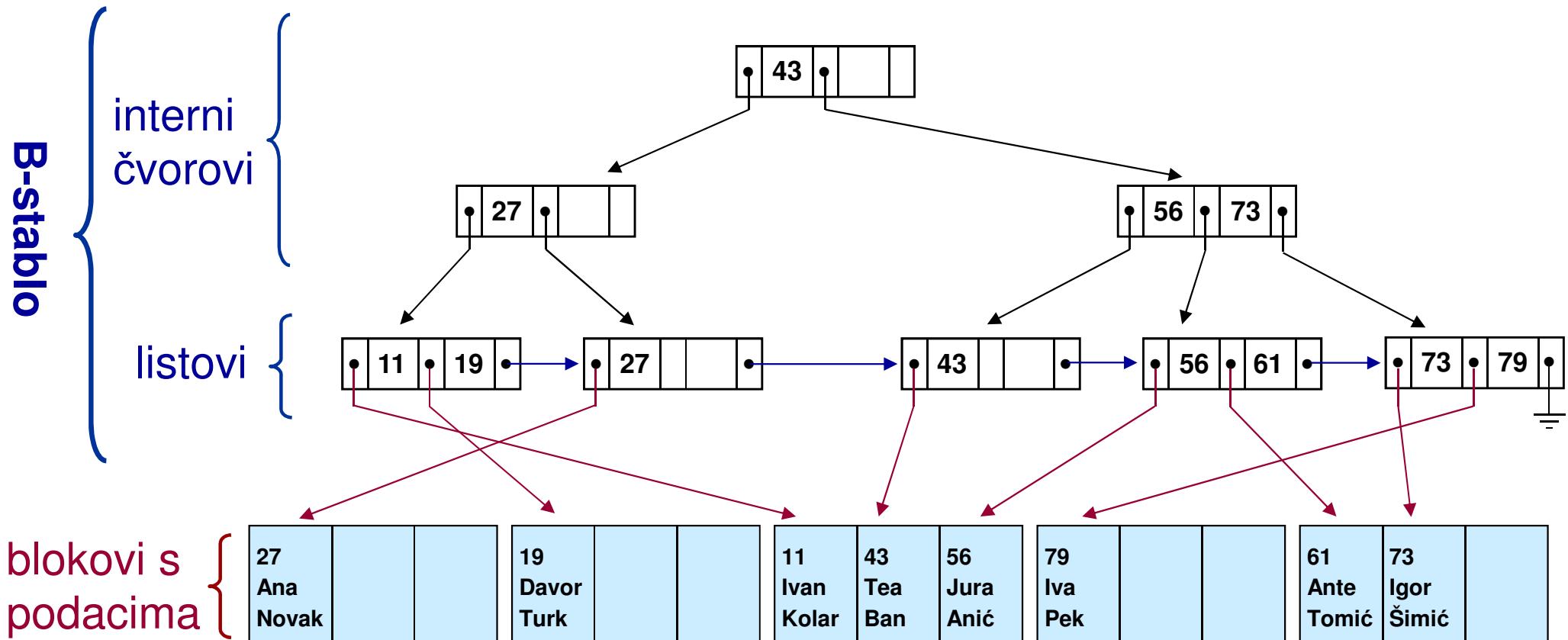


stablo je balansirano



Oznaka **B** u B-stablo znači "**balansirano**"!

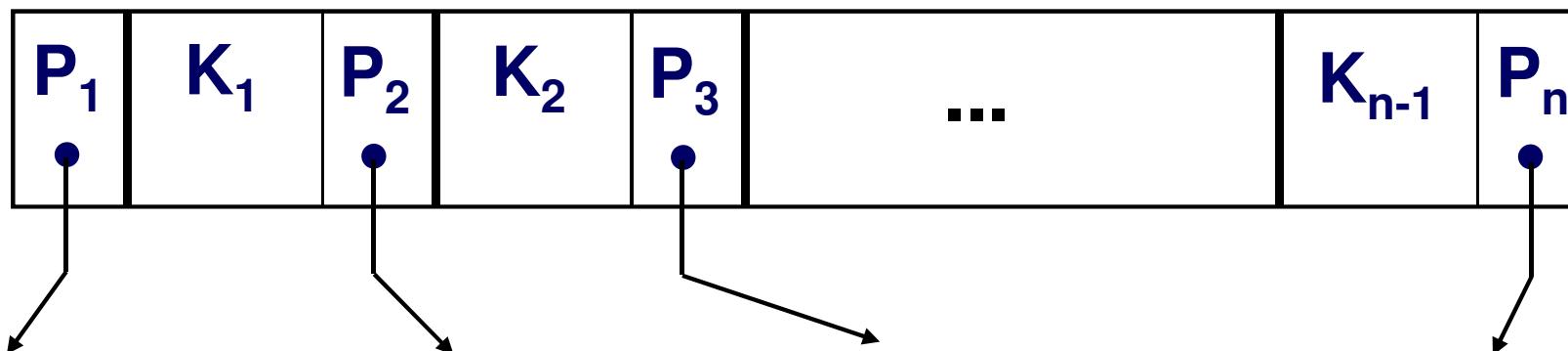
# Struktura B<sup>+</sup>-stabla



- Shema: mbr, ime, prezime; B<sup>+</sup>-stablo je izgrađeno za atribut mbr
- Moguće metode pristupa podacima (za bilo koji ključ pretrage):
  - linearnim pretraživanjem (kao kod neporedane datoteke)
  - ako je ključ pretrage mbr, može se koristiti B-stablo

# Struktura internog čvora B<sup>+</sup>-stabla

- U B<sup>+</sup>-stablu reda  $n$ , **interni čvor** sadrži:
  - najviše  $n$  kazaljki
  - najmanje  $\lceil n/2 \rceil$  kazaljki  $\rightarrow \lceil a \rceil$  je najmanji cijeli broj  $\geq a$ 
    - ovo ograničenje ne vrijedi za korijen (najmanji broj kazaljki je 2)
  - uz  $p$  kazaljki u čvoru, broj pripadnih vrijednosti  $K_i$  u čvoru je  $p-1$ 
    - $K_i$  je vrijednost ključa



podstablo s vrijednostima ključa  $k < K_1$

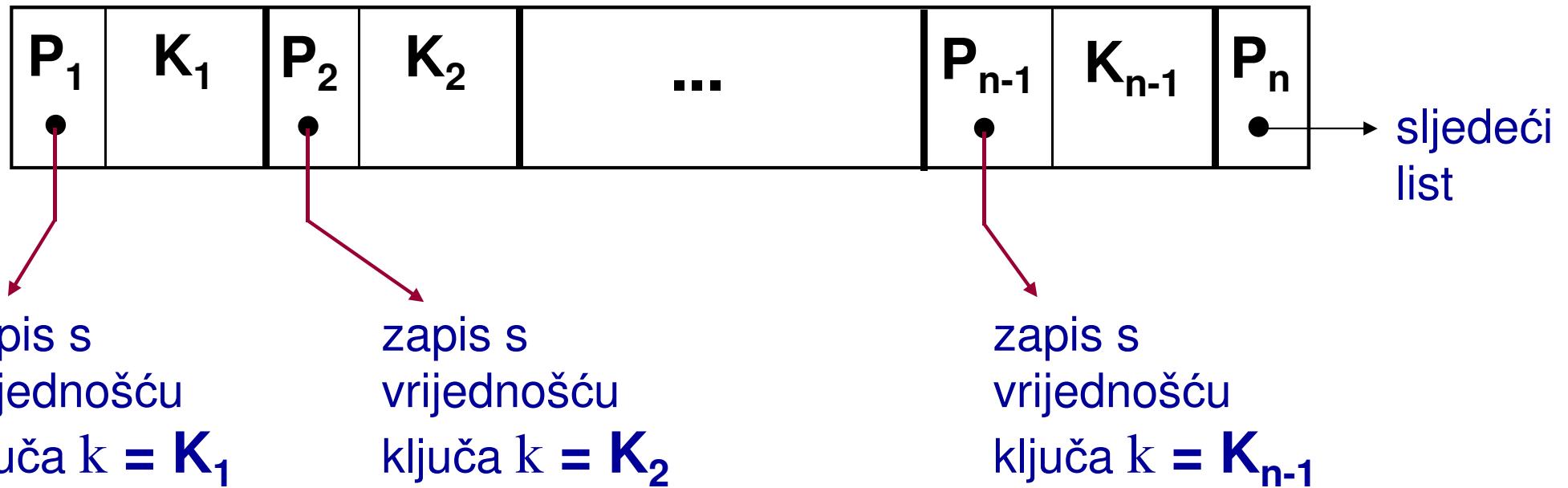
podstablo s vrijednostima ključa  $k$   
 $K_1 \leq k < K_2$

podstablo s vrijednostima ključa  $k$   
 $K_2 \leq k < K_3$

podstablo s vrijednostima ključa  $k \geq K_{n-1}$

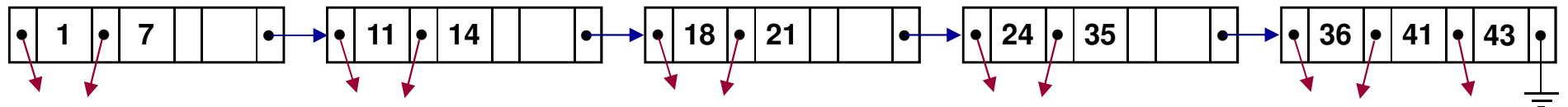
# Struktura lista B<sup>+</sup>-stabla

- U B<sup>+</sup>-stablu reda  $n$ , **list** sadrži:
  - najviše  $n-1$  vrijednosti  $K_i$  i pripadnih kazaljki na zapise
  - najmanje  $\lceil(n-1)/2\rceil$  vrijednosti  $K_i$  i pripadnih kazaljki na zapise
  - svi listovi sadrže kazaljku na sljedeći list
    - omogućava upite tipa od-do (prema zadanim granicama intervala)

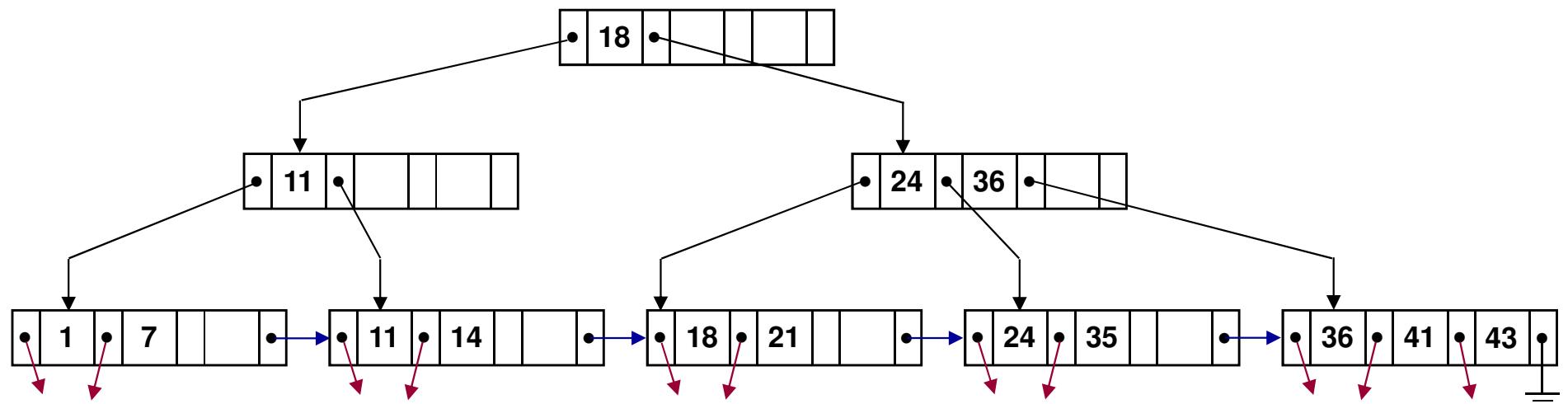


# Zadatak 1.

- Relacija *stud* (*mbr*, *prez*, *ime*) sadrži n-torke sa sljedećim vrijednostima atributa *mbr* : 1, 7, 11, 14, 18, 21, 24, 35, 36, 41, 43. Nacrtati B<sup>+</sup>-stablo reda 4 za atribut *mbr* tako da popunjeno stablo bude minimalna.
- min. broj kazaljki na zapise (n-torke) u jednom listu je  $\lceil (4 - 1) / 2 \rceil = 2$

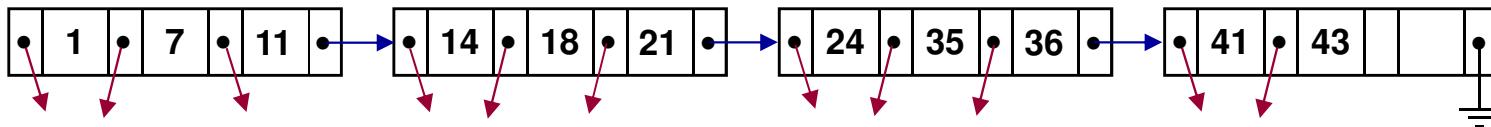


- min. broj kazaljki u jednom internom čvoru (osim korijena) je  $\lceil 4 / 2 \rceil = 2$

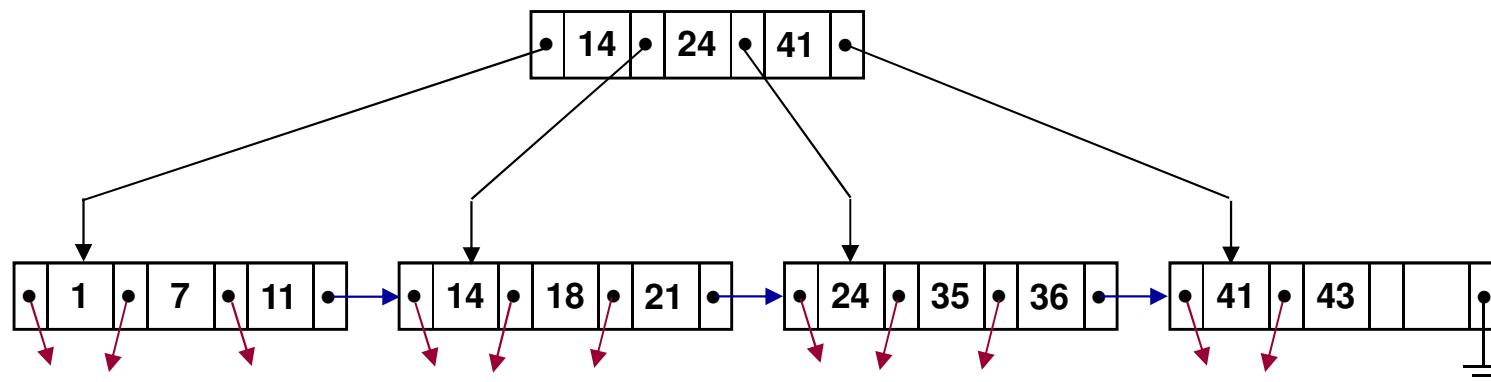


## Zadatak 2.

- Relacija *stud* (*mbr*, *prez*, *ime*) sadrži n-torce sa sljedećim vrijednostima atributa *mbr* : 1, 7, 11, 14, 18, 21, 24, 35, 36, 41, 43. Nacrtati B<sup>+</sup>-stablo reda 4 za atribut *mbr* tako da popunjenošć čvorova u stablu bude maksimalna.
- maksimalni broj kazaljki na zapise (n-torce) u jednom listu je  $4 - 1 = 3$



- maksimalni broj kazaljki u jednom internom čvoru je 4



## Zadatak 3.

- Koliko n-torki sadrži relacija ako je nad njom izgrađeno  $B^+$ -stablo reda 101, s ukupno 5 razina, s minimalno dopuštenom popunjenošću **svih** čvorova
- min. broj kazaljki u jednom listu je  $\lceil (101 - 1) / 2 \rceil = 50$
- min. broj kazaljki u jednom internom čvoru (osim korijena) je  $\lceil 101 / 2 \rceil = 51$
- min. broj kazaljki u korijenu je 2
- relacija sadrži  $2 \cdot 51 \cdot 51 \cdot 51 \cdot 50 \approx 1.33 \cdot 10^7$  n-torki
- **ZAKLJUČAK:** ako je B-stablo reda 101, do svake n-torke u relaciji koja sadrži  $\approx 1.33 \cdot 10^7$  n-torki može se pristupiti, u najlošijem slučaju, korištenjem tek 6 U/I operacija (5 U/I za dohvati lista, 1 U/I za dohvati fizičkog bloka u kojem se nalazi n-torka)

# SQL: Indeksi

## 7. CREATE INDEX Statement

```
CREATE [UNIQUE] INDEX — index — ON [table | synonym] ([column | ASC | DESC])
```

- Obavljanjem naredbe za kreiranje indeksa nad relacijom, nad blokovima s podacima relacije formira se struktura B-stabla

1

```
CREATE TABLE osoba (
 mbr INTEGER
, ime NCHAR(20)
, prez NCHAR(20));

INSERT INTO ...;
```



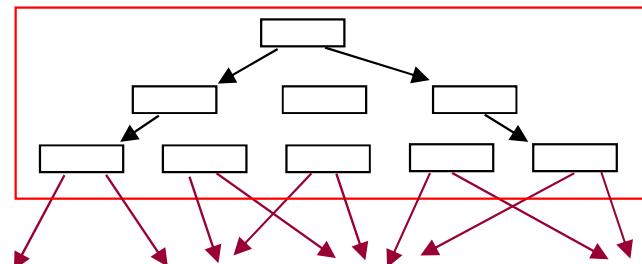
|                    |                  |                    |                  |  |  |                     |                     |  |
|--------------------|------------------|--------------------|------------------|--|--|---------------------|---------------------|--|
| 7<br>Ivan<br>Kolar | 17<br>Tea<br>Ban | 39<br>Jura<br>Anić | 48<br>Iva<br>Pek |  |  | 93<br>Ante<br>Tomić | 67<br>Igor<br>Šimić |  |
|--------------------|------------------|--------------------|------------------|--|--|---------------------|---------------------|--|

2

```
CREATE INDEX osoba_pres
ON osoba (pres);
```



B-stablo za osoba.pres



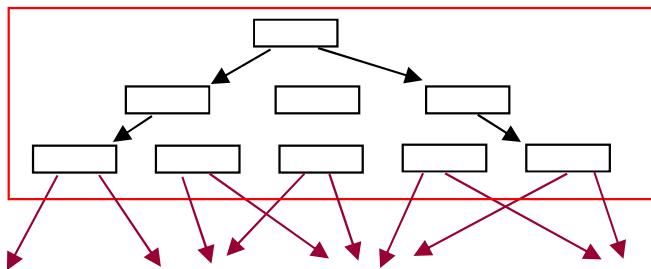
# SQL: Indeksi

- Kreiranjem indeksa uz navođenje rezervirane riječi UNIQUE osigurava se jedinstvenost vrijednosti navedenog atributa

```
CREATE UNIQUE INDEX osoba_mbr
ON osoba (mbr);
```



B-stablo za osoba.mbr



|                    |                  |                    |                  |  |  |                     |                     |  |
|--------------------|------------------|--------------------|------------------|--|--|---------------------|---------------------|--|
| 7<br>Ivan<br>Kolar | 17<br>Tea<br>Ban | 39<br>Jura<br>Anić | 48<br>Iva<br>Pek |  |  | 93<br>Ante<br>Tomić | 67<br>Igor<br>Šimić |  |
|--------------------|------------------|--------------------|------------------|--|--|---------------------|---------------------|--|

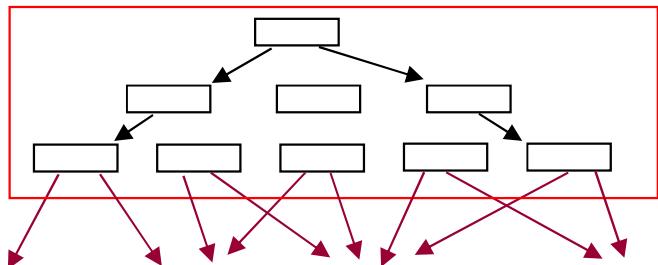
- ukoliko se indeks pokuša kreirati nad relacijom u kojoj već postoje duplikati vrijednosti atributa mbr, sustav će odbiti kreirati indeks i dojaviti pogrešku
- pokuša li se nakon kreiranja ovog indeksa unijeti n-torka s vrijednošću atributa mbr koja već postoji u nekoj n-torci, sustav će odbiti operaciju i dojaviti pogrešku

```
DROP INDEX osoba_mbr;
```

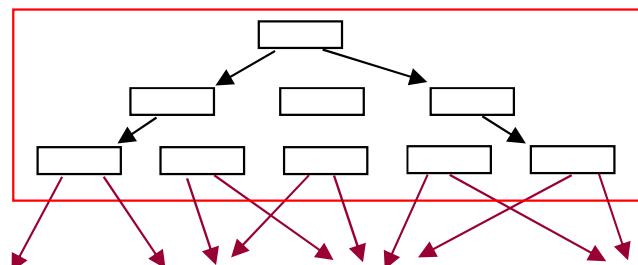
# Više indeksa nad istom relacijom

- nad istom relacijom može se izgraditi više indeksa

B-stablo za osoba.mbr



B-stablo za osoba.prez



|                     |                     |                    |                  |  |  |  |
|---------------------|---------------------|--------------------|------------------|--|--|--|
| 11<br>Ivan<br>Kolar | 43<br>Tea<br>Ban    | 56<br>Jura<br>Anić | 79<br>Iva<br>Pek |  |  |  |
| 61<br>Ante<br>Tomić | 73<br>Igor<br>Šimić |                    |                  |  |  |  |

- B-stabla (indeksi) prikazani u primjeru omogućuju efikasno obavljanje upita s uvjetima ( $=$ ,  $>$ ,  $\geq$ ,  $<$ ,  $\leq$ , BETWEEN) i efikasno sortiranje (ASC, DESC)
  - za atribut mbr
  - za atribut prez
- prema uvjetima koji sadrže atribut ime, podacima se može pristupati jedino linearnom pretragom svih blokova

# Za koje atribute treba kreirati indeks?

- Primjer:

```
CREATE TABLE stud (
 mbr INTEGER
, ime NCHAR(20)
, prez NCHAR(20));
```

- Često se postavljaju upiti oblika:

```
SELECT * FROM stud
WHERE mbr = 12345;
```

```
SELECT * FROM stud
WHERE prez > 'Kolar'
ORDER BY prez;
```

⇒ Kreirati indeks za mbr i indeks za prez

- Što se nakon kreiranja navedenih indeksa dešava pri obavljanju:

```
UPDATE stud SET prez = UPPER(prez)
WHERE ime = 'Ivan';
```

- n-torce se pronalaze linearnom pretragom (loše), a zbog izmjene vrijednosti atributa prez mora se izmijeniti sadržaj B-stabla za indeks nad atributom prez (loše)

```
UPDATE stud SET ime = UPPER(ime)
WHERE prez = 'Horvat';
```

- n-torce se pronalaze pomoću B-stabla (dobro), nad atributom ime nije izgrađen indeks, stoga ne postoji B-stablo čiji se sadržaj mora izmijeniti (dobro)

# Za koje atributе ne treba kreirati indeks?

---

- ako vrijednosti atributa imaju relativno mali broj različitih vrijednosti
  - npr. atribut spolOsobe s dozvoljenim vrijednostima M, Ž
- ako relaciji predstoji velik broj upisa, izmjena ili brisanja n-torki. Preporuča se u takvim slučajevima postojeće indekse izbrisati, te ih ponovo izgraditi tek nakon obavljenih promjena nad podacima
- ako relacija sadrži relativno mali broj n-torki (sve n-torke su pohranjene u nekoliko blokova). U takvim slučajevima B-stablo ne pridonosi efikasnosti pretrage
  - npr. relacija zupanija

# Složeni indeksi

```
CREATE TABLE stud (
 mbr INTEGER
, ime NCHAR(20)
, prez NCHAR(20));
```

```
CREATE INDEX stud_ime ON stud (ime);
CREATE INDEX stud_pres ON stud (pres);
```

- efikasno se obavljaju upiti oblika:

```
SELECT * FROM stud
WHERE prez = 'Kolar';
```

```
SELECT * FROM stud
WHERE ime = 'Ivan';
```

- upit oblika:

```
SELECT * FROM stud
WHERE prez = 'Kolar'
AND ime = 'Ivan';
```

- pomoću indeksa nad atributom prez dohvatić će se n-torce studenata čije je prezime 'Horvat', ali će se u dobivenom skupu n-torki linearnom pretragom morati pronaći oni čije je ime 'Ivan' (ili indeksom po imenu, a onda linearno po prezimenu)

# Složeni indeksi

- Prethodni upit se efikasnije obavlja ako se umjesto posebnih indeksa za atributime ime i prezime, kreira složeni indeks:

```
CREATE INDEX stud_pres_ime ON stud (pres, ime);
```

- problem: ovaj indeks se koristi za upite oblika:

```
SELECT * FROM stud
WHERE prez = 'Kolar'
AND ime = 'Ivan';
```

=

```
SELECT * FROM stud
WHERE ime = 'Ivan'
AND prez = 'Kolar';
```

- također i za upite oblika:

```
SELECT * FROM stud
WHERE prez = 'Kolar';
```

- ali se ne može koristiti za upite oblika:

```
SELECT * FROM stud
WHERE ime = 'Ivan';
```

# Složeni indeksi

- Kako upotreba složenih indeksa utječe na sortiranje:

```
CREATE INDEX stud_pres_ime1 ON stud (pres, ime);
```

- indeks osoba\_pres\_ime1 se efikasno koristi za sortiranje oblika:

```
SELECT * FROM stud
ORDER BY prez, ime;
```

```
SELECT * FROM stud
ORDER BY prez DESC, ime DESC;
```

- ali ne i za:

```
SELECT * FROM stud
ORDER BY prez DESC, ime;
```

- ako se kreira indeks:

```
CREATE INDEX stud_pres_ime2 ON stud (pres DESC, ime);
```

- indeks osoba\_pres\_ime2 se efikasno koristi za sortiranje oblika:

```
SELECT * FROM stud
ORDER BY prez DESC, ime;
```

```
SELECT * FROM stud
ORDER BY prez, ime DESC;
```

# Složeni indeksi

- Ako je nad relacijom r (ABCD) kreiran složeni indeks r\_abc1

```
CREATE INDEX r_abc1 ON r (A, B, C);
```

tada sljedeće indekse nije potrebno kreirati:

```
CREATE INDEX r_abc2 ON r (A DESC, B DESC, C DESC);
CREATE INDEX r_a1 ON r (A);
CREATE INDEX r_a2 ON r (A DESC);
CREATE INDEX r_ab1 ON r (A, B);
CREATE INDEX r_ab2 ON r (A DESC, B DESC);
```

- Indekse r\_abc2, r\_a1, r\_a2, r\_ab1 i r\_ab2 ne treba kreirati jer SUBP može koristiti indeks r\_abc1 u svim slučajevima u kojima bi se koristili indeksi r\_abc2, r\_a1, r\_a2, r\_ab1 i r\_ab2.

## Zadatak 5. zadana je relacija stud (mbr ime prez pbrStan)

---

- Za relaciju stud kreirati najmanji mogući broj indeksa koji će omogućiti efikasno obavljanje (pomoću B<sup>+</sup>-stabla) svih navedenih upita:
    1. SELECT \* FROM stud ORDER BY ime DESC, prez;
    2. SELECT \* FROM stud ORDER BY ime DESC, prez DESC;
    3. SELECT \* FROM stud ORDER BY ime, prez, pbrStan;
    4. SELECT \* FROM stud WHERE prez = 'Novak' AND ime = 'Ivo';
    5. SELECT \* FROM stud WHERE pbrStan > 51000 ORDER BY pbrStan DESC;
- 

1. (ime DESC, prez)
2. (ime DESC, prez DESC)
3. (ime, prez, pbrStan) - ali sada više nije potreban indeks pod 2.
4. može se koristiti indeks pod 3.
5. (pbrStan)

**Konačno rješenje** - kreirati indekse za:

(ime DESC, prez)  
(ime, prez, pbrStan)  
(pbrStan)

SQL naredbe za  
kreiranje indeksa  
napisati za vježbu!