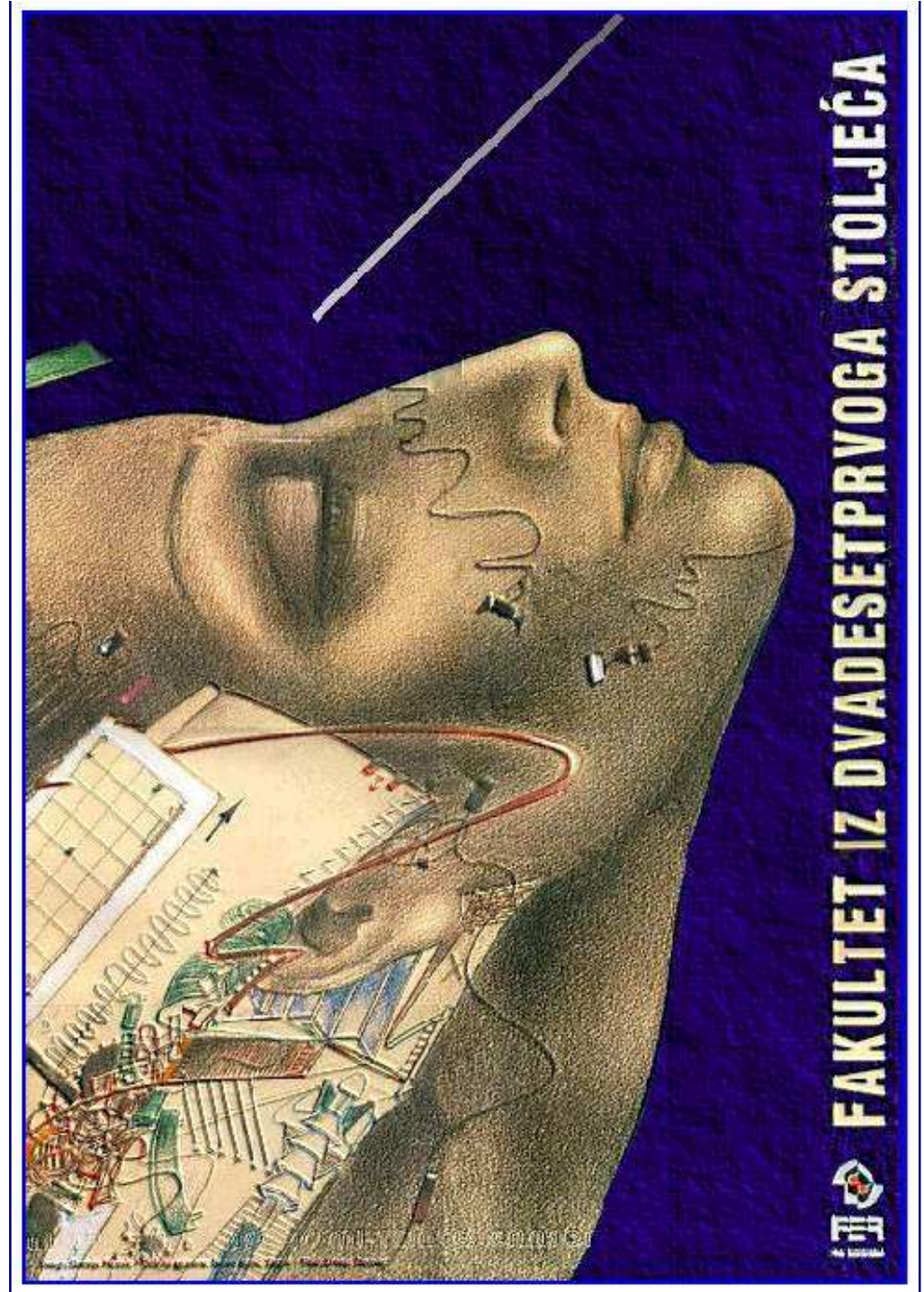


Baze podataka

svibanj 2009.

Priprema za 2MI



Analiza prethodnih međuispita

- 1. MI – SQL + relacijska algebra
- SQL
 - EXISTS
 - UPDATE
 - DELETE
- Integritetska ograničenja
- Funkcijske zavisnosti
- B-stablo
- Indeksi
- Normalizacija

Model – ocjenjivanje serija

Model sadrži podatke o serijama, korisnicima i ocjenama koje su korisnici dali pojedinoj seriji.

serija

sifSerija	naziv	brojSezona
1	Reba	6
2	Desperate Housewives	5
3	Will & Grace	8
4	Allo Allo	9
5	Ghostwhisperer	5
...

korisnik

sifKorisnik	ime	prezime	ocjenaAktivnost
1	Ivana	Klarić	5
2	Ivana	Balić	2
4	Luka	Klarić	1
5	Krešimir	Peroš	3
19	Robert	Britvić	3
...

ocjenaSerija

sifSerija	sifKorisnik	datum	ocjena
1	2	2.2.2009.	5,00
5	10	1.10.2008.	7,5
15	19	12.1.2009.	2,5
16	1	10.10.2008.	8
...

1. zadatak

- Svim korisnicima koji nisu u posljednjih godinu dana ocijenili nijednu seriju smanjiti ocjenu aktivnosti za 1.

```
UPDATE korisnik SET ocjenaAktivnost = ocjenaAktivnost - 1
WHERE sifKorisnik NOT IN
    (SELECT DISTINCT sifKorisnik FROM ocjenaSerija
     WHERE datum > MDY (MONTH(TODAY), DAY(TODAY), YEAR(TODAY) - 1))
```

2. zadatak

- Obrisati podatke o korisnicima koji su ocijenili manje od 10 serija i dali svima ocjenu manju od 5.

```
DELETE FROM korisnik
WHERE (SELECT COUNT(*) FROM ocjenaSerija
      WHERE ocjenaSerija.sifKorisnik = korisnik.sifKorisnik) < 10
AND NOT EXISTS (SELECT * FROM ocjenaSerija
                WHERE ocjenaSerija.sifKorisnik = korisnik.sifKorisnik
                AND ocjena >= 5)
```

3. zadatak

- Za svaku seriju koja se prikazivala dulje od 5 sezona, ispisati naziv, broj sezona, godinu ocjenjivanja i prosječnu ocjenu koju je dobila te godine od strane korisnika. U ispisu se trebaju pojaviti i serije koje nisu bile nijednom ocijenjene (a prikazivale su se dulje od 5 sezona). Zapise poredati silazno prema prosječnoj ocjeni i godini.

```
SELECT naziv, brojSezona, YEAR(datum) AS godina
      , AVG (ocjena) AS prosjek
FROM serija LEFT JOIN ocjenaSerija
      ON serija.sifSerija = ocjenaSerija.sifSerija
WHERE serija.brojSezona > 5
GROUP BY naziv, brojSezona, 3
ORDER BY prosjek DESC, godina DESC
```

4. Zadatak (a)

- Kreirati privremenu relaciju `top10` koja će sadržavati naziv serije i prosjek za 10 najbolje plasiranih serija.

```
CREATE TEMP TABLE top10 (  
  naziv CHAR(30),  
  prosjek DECIMAL(4,2));
```

```
INSERT INTO top10  
SELECT FIRST 10 naziv, AVG(ocjena) AS prosjek  
  FROM serija INNER JOIN ocjenaSerija  
    ON serija.sifSerija = ocjenaSerija.sifSerija  
 GROUP BY serija.sifSerija, naziv  
 ORDER BY 2 DESC
```

ili

```
SELECT FIRST 10 naziv, AVG(ocjena) AS prosjek  
  FROM serija INNER JOIN ocjenaSerija  
    ON serija.sifSerija = ocjenaSerija.sifSerija  
 GROUP BY serija.sifSerija, naziv  
 ORDER BY 2 DESC  
 INTO TEMP top10
```

4. Zadatak (b)

- Iz relacije `ocjenaSerija` obrisati podatke o 10 najbolje plasiranih serija.

```
SELECT FIRST 10 serija.sifSerija, AVG(ocjena) AS prosjek
FROM serija INNER JOIN ocjenaSerija
    ON serija.sifSerija = ocjenaSerija.sifSerija
GROUP BY serija.sifSerija
ORDER BY 2 DESC
INTO TEMP top10;
```

```
DELETE FROM ocjenaSerija
WHERE sifSerija IN (SELECT sifSerija FROM top10);
```


5. zadatak

- Kreirati pogled koji će sadržavati najbolje ocjenjenu seriju (ili više njih) za 2005. godinu. Pogled treba imati attribute: godina, sifSerija i naziv. Najbolje ocjenjena serija je ona koja ima najveću prosječnu ocjenu.

```
CREATE VIEW najboljeSerije (godina, sifSerija, naziv)
  SELECT YEAR(datum), naziv
    FROM serija INNER JOIN ocjenaSerija
      ON serija.sifSerija = ocjenaSerija.sifSerija
   WHERE YEAR(datum) = 2005
   GROUP BY 1, ocjenaSerija.sifSerija, naziv
  HAVING AVG(ocjena) >= ALL(SELECT AVG(sveOcjene.ocjena)
                             FROM ocjenaSerija sveOcjene
                            WHERE YEAR(sveOcjene.datum) = 2005
                             GROUP BY sveOcjene.sifSerija)
```

Integritetska ograničenja

Napisati SQL naredbe koje će kreirati relacije radnik i orgjed prema relacijskim shemama:

RADNIK={ sifRadnik, ime, prez, sifOrgJed , datZap, datRodj}

ORGJED={ sifOrgJed, nazOrgJed , sifNadOrgJed}.

Smisleno odaberite tipove podataka. Osigurati entitetski integritet i integritet ključa u obje relacije. Također:

- u relaciji radnik niti jedan atribut ne smije biti NULL
- vrijednost atributa sifRadnik u relaciji radnik mora biti broj veći od 100,000
- datum zaposlenja mora biti barem 18 godina nakon datuma rođenja
- vrijednost atributa sifOrgJed u relaciji radnik mora odgovarati postojećoj org. jedinici u relaciji orgjed
- ako je poznata, vrijednost atributa sifNadOrgJed u relaciji orgjed mora odgovarati postojećoj org. jedinici u relaciji orgjed
- prilikom brisanja zapisa iz relacije orgjed brišu se i svi zapisi iz relacije orgjed koji se podređeni zapisu koji se briše
- u relaciju orgjed nije dozvoljeno unijeti dvije org.jedinice istog naziva

```
CREATE TABLE orgjed (  
    sifOrgJed      INTEGER PRIMARY KEY  
    , nazOrgJed    CHAR(100) NOT NULL  
    , sifNadOrgJed INTEGER  
    , UNIQUE(nazOrgJed)  
    , FOREIGN KEY (sifNadOrgJed) REFERENCES orgjed(sifOrgJed) ON  
      DELETE CASCADE  
);
```

```
CREATE TABLE radnik (  
    sifRadnik INTEGER PRIMARY KEY CHECK(sifRadnik > 100000)  
    , ime      CHAR(20)  NOT NULL  
    , prez     CHAR(30)  NOT NULL  
    , sifOrgJed INTEGER   NOT NULL  
    , datRodj   DATE      NOT NULL  
    , datZap    DATE      NOT NULL  
    , FOREIGN KEY (sifOrgJed) REFERENCES orgjed(sifOrgJed)  
    , CHECK ( YEAR(datZap) - YEAR(datRodj)  > 18)  
);
```

B-stablo

Relacija *orgJed(sifOrgJed, nazOrgJed)* sadrži n-torke sa sljedećim vrijednostima atributa *sifOrgJed*:

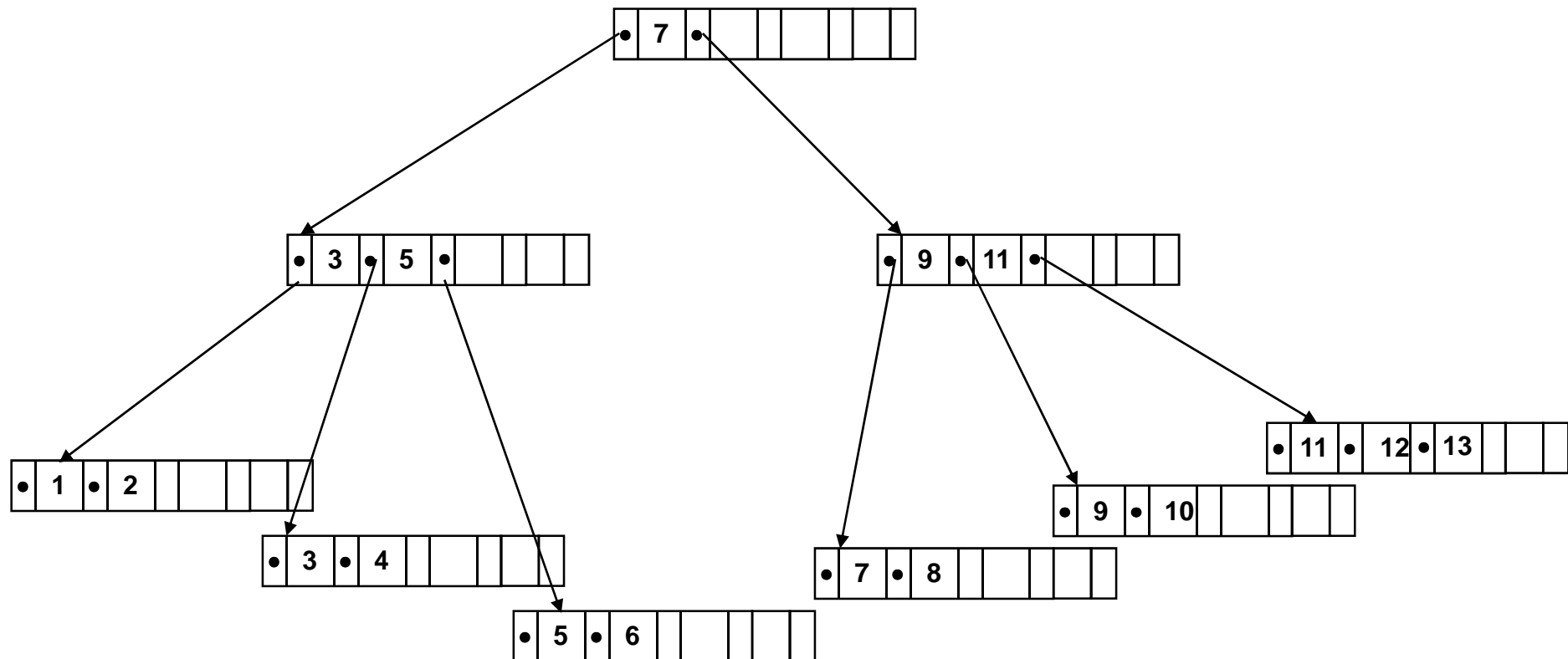
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13.

Nacrtati B⁺-stablo reda **5** za atribut *sifOrgJed* tako da broj UI operacija pri pretraživanju bude:

- a) **maksimalan** (popunjenost čvorova u stablu minimalna)
- b) **minimalan** (popunjenost čvorova u stablu maksimalna)

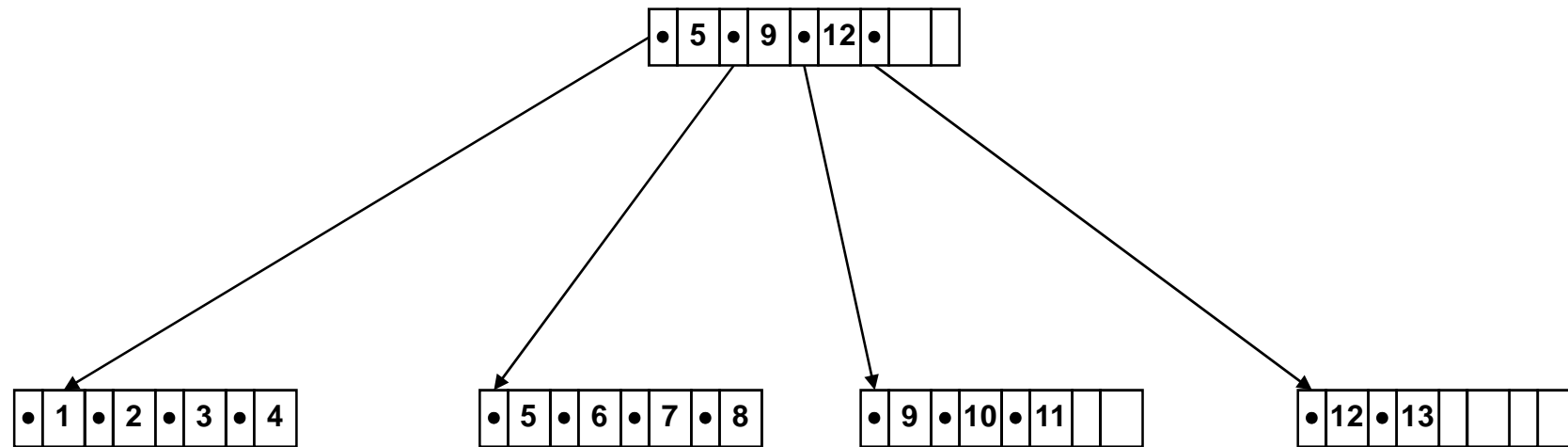
- **interni čvor [3, 5]**
najmanje $\lceil n/2 \rceil$ kazaljki ($\lceil a \rceil$ je najmanji cijeli broj $\geq a$)
najviše n kazaljki
- **list [2, 4]**
najmanje $\lceil (n-1)/2 \rceil$ K_i i pripadnih kazaljki na zapise
najviše $n-1$
- **korijen [2, 5]**
najmanji broj kazaljki je 2
najviše n kazaljki (kao interni čvor)

a) maksimalan (popunjenost čvorova u stablu minimalna)



Koliko UI operacija?

b) minimalan (popunjenost čvorova u stablu maksimalna)



Koliko UI operacija?

Indeksi

Zadana je relacija **radnik(sifRadnik, ime, prez, sifOrgJed)**. Napisati SQL naredbe za kreiranje najmanjeg mogućeg broja indeksa koji će omogućiti efikasno obavljanje (pomoću B+ stabla) navedenih upita.

1. `SELECT * FROM radnik WHERE sifOrgJed = 13;`
2. `SELECT * FROM radnik WHERE ime = 'Marko' AND prezime = 'Horvat';`
3. `SELECT * FROM radnik ORDER BY prezime, ime DESC;`
4. `SELECT * FROM radnik ORDER BY prezime, ime, sifOrgJed;`
5. `SELECT * FROM radnik ORDER BY prezime DESC, ime DESC;`

Napišite jednu SELECT naredbu nad relacijom nastavnik koja se NE MOŽE efikasno obaviti kreiranim indeksima. Naredba mora u WHERE dijelu koristiti barem jedan, a u ORDER BY dijelu barem dva atributa.

```
CREATE INDEX i1 ON radnik (sifOrgJed);           -- 1
CREATE INDEX i2 ON radnik(prezime, ime, sifOrgJed); -- 4, 5, 2
CREATE INDEX i3 ON radnik(prezime, ime DESC);    -- 3
```

```
SELECT * FROM radnik
WHERE ime = 'Krešimir'
ORDER BY sifOrgJed, ime
```

Normalizacija

Fitness centar evidentira raspored svojih treninga.

Relacijska shema **FCRASPORED** sastoji se od sljedećih atributa:

- **rbrDanUTjednu** – redni broj dana u tjednu (0-nedjelja, 1-ponedjeljak,...)
- **sifDvorana** – šifra dvorane u kojoj se trening održava
- **kapDvorana** – kapacitet dvorane u kojoj se trening održava
- **vrijPoc** – vrijeme početka
- **sifVrTrening** – šifra vrste treninga
- **nazVrTrening** – naziv vrste treninga (Step, Pilates, Booty Express, ...)
- **trajeTrening** – trajanje treninga
- **sifTrener** – šifra trenera
- **imeTrener** – ime trenera
- **prezimeTrener** – prezime trenera

Normalizacija

Odaberite ključ relacijske sheme **FCRASPORED** tako da ona bude u 1NF, a zatim postupno normalizirajte relacijsku shemu na 2NF i 3NF ako vrijedi:

- istog dana u isto vrijeme može se održavati više treninga ali ne u istoj dvorani
- svi treninzi iste vrste jednako traju (npr. Booty Express uvijek traje 30 min; Step uvijek traje 60 min)
- u određenom terminu (dan u tjednu + vrijeme u danu) u jednoj dvorani održava se samo jedna vrsta treninga
- u istom terminu se u različitim dvoranama može održavati ista vrsta treninga
- u istoj dvorani i istom terminu trening uvijek drži isti trener (npr. srijedom u 17:00 u dvorani D01 trening uvijek drži trenerica Sanja Antić a utorkom u 18:00 u dvorani D02 trener Saša Grubišić)


Normalizacija – 1NF

FCRASPORED = {rbrDanUTjednu, sifDvorana, kapDvorana, vrijPoc,
sifVrTrening, nazVrTrening, trajeTrening,
sifTrener, imeTrener, prezimeTrener}

$K_{FCRASPORED} = \{rbrDanUTjednu, sifDvorana, vrijPoc\}$

Normalizacija – 2NF

2NF? FCRASPORED = {rbrDanUTjednu, sifDvorana, kapDvorana, vrijPoc,
sifVrTrening, nazVrTrening, trajeTrening,
sifTrener, imeTrener, prezimeTrener}



$K_{FCRASPORED} = \{rbrDanUTjednu, sifDvorana, vrijPoc\}$

Normalizirati relacijsku shemu FCRASPORED na 2NF.

DVORANA = { sifDvorana, kapDvorana } $K_{DVORANA} = \{ sifDvorana \}$

$FCRASPORED_1 = \{rbrDanUTjednu, sifDvorana, vrijPoc,$
sifVrTrening, nazVrTrening, trajeTrening,
sifTrener, imeTrener, prezimeTrener}

$K_{FCRASPORED_1} = \{rbrDanUTjednu, sifDvorana, vrijPoc\}$

Normalizacija – 3NF

3NF? DVORANA= {sifDvorana, kapDvorana} $K_{DVORANA} = \{ \text{sifDvorana} \}$ 3NF OK.

3NF? FCRASPORED₁= {rbrDanUTjednu, sifDvorana, vrijPoc,
sifVrTrening, nazVrTrening, trajeTrening,
sifTrener, imeTrener, prezimeTrener}



$K_{FCRASPORED1} = \{ \text{rbrDanUTjednu, sifDvorana, vrijemePoc} \}$

Normalizirati relacijsku shemu FCRASPORED₁ na 3NF.

VRTRENING= {sifVrTrening, nazVrTrening, trajeTrening} $K_{VRTRENING} = \{ \text{sifVrTrening} \}$

TRENER = {sifTrener, imeTrener, prezimeTrener} $K_{TRENER} = \{ \text{sifTrener} \}$

FCRASPORED₂= {rbrDanUTjednu, sifDvorana, vrijPoc, sifVrTrening, sifTrener}

$K_{FCRASPORED2} = \{ \text{rbrDanUTjednu, sifDvorana, vrijemePoc} \}$

Normalizacija – 3NF

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

FCRASPORED₂, DVORANA, VRTRENING i TRENER

Primjer korištenja pravila o akumulaciji

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

dokazati vrijedi li 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) $\Rightarrow MQ \rightarrow MQR P$
- ne postoji FZ kojom bi se moglo nastaviti "uvećavati desnu stranu"
- $\Rightarrow MQ \rightarrow LN$ ne vrijedi