

šalabahter

fz

funkcijske zavisnosti

### 1. REFLEKSIVNOST

uzmi lijevu stranu zadane funkcijske zavisnosti, napiši strelicu, i opet zapiši lijevu stranu fz.

za fz  $AB \rightarrow CD$  treba napisat:  $AB \rightarrow AB$  (refleksivnost)

### 2. AKUMULACIJA

gledaj desnu stranu svoje najnovije fz, vidi jel neki skup atributa s desna ima neku definiranu fz u zadatku

ako ima, prepisi slove s desne strane te fz u desnu stranu tvoje fz

za tvoju novu fz  $AB \rightarrow ABC$  i njihovu  $BC \rightarrow D$  treba napisat

$AB \rightarrow ABC$  i  $BC \rightarrow D \Rightarrow AB \rightarrow ABCD$  (akumulacija)

ako si nešto dodala novo, ispočetka idi na akumulaciju, ako nisi idi na korak 3

### 3. DEKOMPOZICIJA

makni s desna sve stvari koje ti ne trebaju

za tvoju novu fz  $AB \rightarrow ABCDEF$ , a trebaš dokazat  $AB \rightarrow EF$

treba napisat:  $AB \rightarrow ABCDEF \Rightarrow AB \rightarrow EF$  (dekompozicija)

ako to ne možeš dobit nikako, to je dovoljan dokaz da je nema, napiši nešto lijepo o tome

npr. dani skup fz ne omogućuje da se EF pojavi na desnoj strani dakle fz ne vrijedi

---

pronalazak mogućeg ključa

isti postupak kao i dokazivanje fz, treba doći do neposredno prije 3. koraka,

ali ga treba provoditi za: **zadani mogući ključ** (npr CDE), i za **sve njegove podskupove** (CD, DE, CE, C, D, E) - nikad neće biti tak puno

i sad - ključ je moguć ako vrijedi

za mogući ključ desno imamo sve preostale atribute (npr došli smo do CDE -> ABCDEF)

i za sve podskupove ista stvar ne vrijedi (nije se dogodilo tipa C -> ABCDEF)

---

Joinanje dvije tablice iz vennovog dijagrama

<https://sql-joins.leopard.in.ua/>

---

create

```
CREATE TABLE mjesto (  
  pbr INTEGER NOT NULL,  
  nazMjesto VARCHAR(30) NOT NULL,  
  glGrad VARCHAR(1) DEFAULT 'N'  
  sifZup SMALLINT  
);
```

```
CREATE TABLE tableName (  
  Int1 INTEGER NOT NULL,  
  Int2 SMALLINT  
  String1 VARCHAR(length) NOT NULL,  
  String2 VARCHAR(length) DEFAULT 'VALUE'  
  serijskiBroj SERIAL  
);
```

insert

```
INSERT INTO mjesto (  
  pbr, mbr, pbrStan)  
VALUES (  
  'Kolar', 101, 10000);
```

```
INSERT INTO nazivTablice (  
  atr1, atr3, atr2)  
VALUES (  
  'value1', value3, value2);
```

\*u ovoj situaciji ne moramo uopće spominjati one atribute koje želimo da budu NULL

\*kao niti atribute koji su SERIAL (oni se povećavaju sami od sebe i za njih nas nije briga)

```
INSERT INTO polozioFiz
SELECT stud.mbr, ime, prez //atributi moraju biti poslagani dobrim redosljedom...
FROM stud, ispit          //ali se ne moraju poklapati s nazivima u gornjoj
WHERE stud.mbr = ispit.mbr
AND predmet = 'Fizika'
AND ocjena > 1;
```

\*uzimanje iz jedne tablice i stavljanje u drugu

## Delete

```
DELETE FROM mjesto
WHERE sifZup = 7;
```

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

\*ako želiš koristit podupit

## Update

```
UPDATE ispit
SET ocjena = ocjena + 1,
    brBod = NULL
WHERE ocjena < 5;
```

```

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;

```

\*ako ima puno slučajeva

```

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)
);

```

\*podupit u podupitu u UPDATE-u