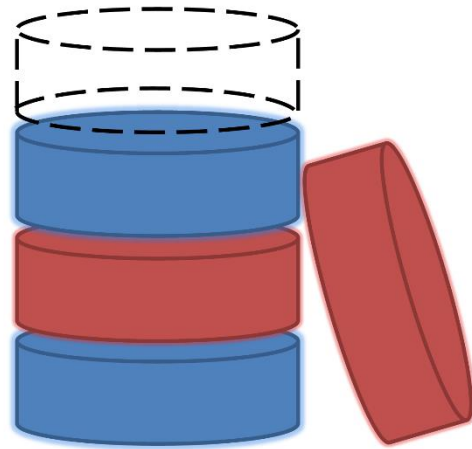
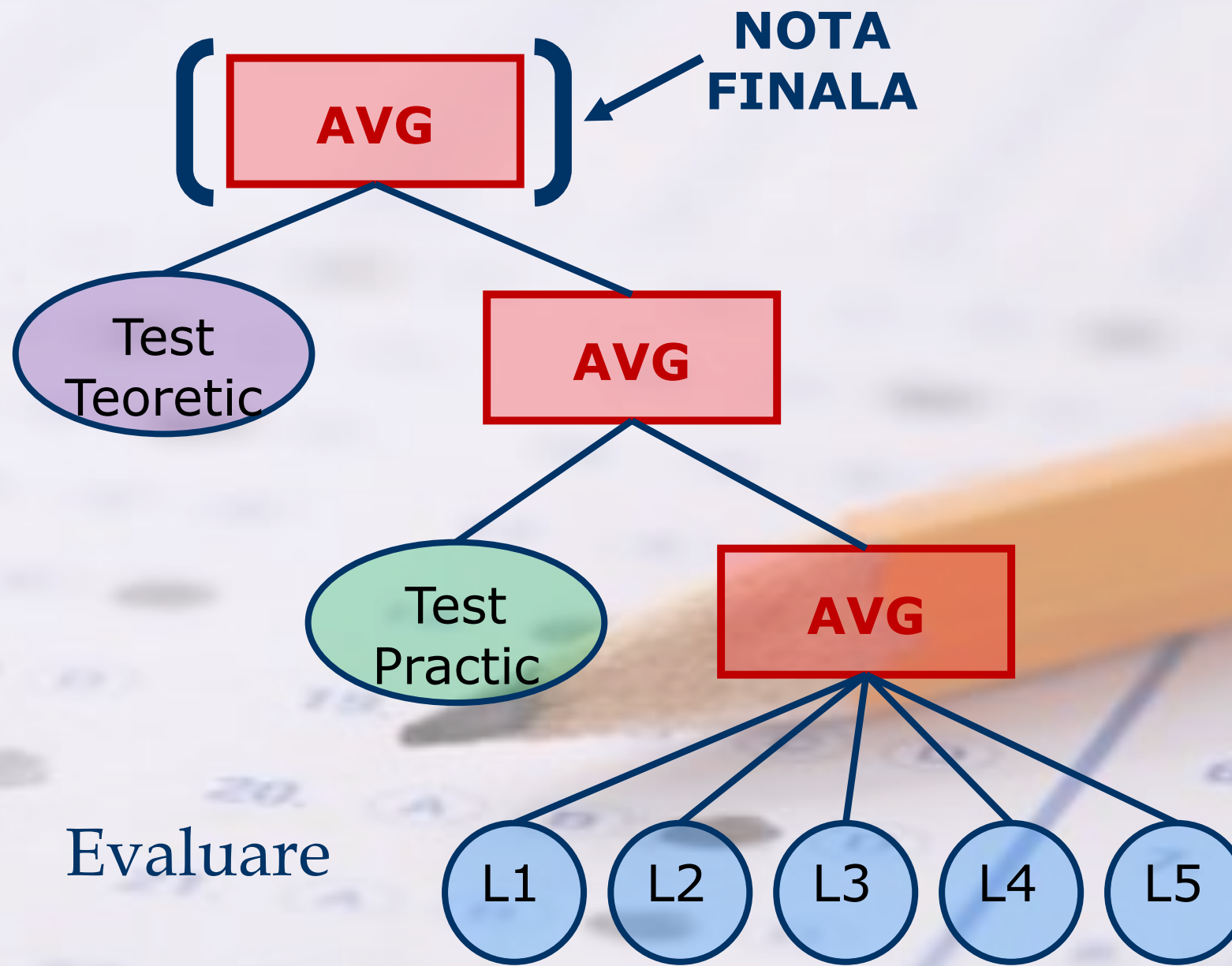


# Probleme






# Test teoretic

- Test grilă: 20 întrebări cu 4 răspunsuri posibile  
– un singur răspuns corect. (20%)
- 2-3 probleme (30%)

# Cel puțin 5 întrebări din testul grilă au structura:

Următoarele interogări SQL se execută pe relațiile  $R(a; b; c)$  și  $S(c; d; e)$

DISTINCT

$Q_1$ : SELECT  b, d FROM R  
INNER JOIN S ON R.c = S.c;

$Q_2$ : SELECT d, b FROM R, S  
WHERE R.c = S.c;  
ORDER BY d

- A.  $Q_1$  și  $Q_2$  produc același răspuns
- B. Răspunsul lui  $Q_1$  este întotdeauna conținut în răspunsul lui  $Q_2$
- C. Răspunsul lui  $Q_2$  este întotdeauna conținut în răspunsul lui  $Q_1$
- D.  $Q_1$  și  $Q_2$  produc răspunsuri diferite.

# Problema 1

Fie următoarea instanță a relației *Studenti* :

sid	nume	email	varsta	nota
2833	Jones	<a href="mailto:jones@scs.ubbcluj.ro">jones@scs.ubbcluj.ro</a>	19	9
2877	Smith	<a href="mailto:smith@scs.ubbcluj.ro">smith@scs.ubbcluj.ro</a>	20	8
2976	Jones	<a href="mailto:jones@math.ubbcluj.ro">jones@math.ubbcluj.ro</a>	21	10
2765	Mary	<a href="mailto:mary@math.ubbcluj.ro">mary@math.ubbcluj.ro</a>	22	7.7
3000	Dave	<a href="mailto:dave@cs.ubbcluj.ro">dave@cs.ubbcluj.ro</a>	18	5.5
3010	Smith	<a href="mailto:smith2@scs.ubbcluj.ro">smith2@scs.ubbcluj.ro</a>	20	7
3020	Sam	<a href="mailto:sam@scs.ubbcluj.ro">sam@scs.ubbcluj.ro</a>	19	9.5

1. Dați exemplu de un atribut (sau mulțime de attribute) ce nu poate fi cheie candidat, considerând că instanța de mai sus este legală.
2. Putem deduce dacă un atribut (sau mulțime de attribute) este cheie candidat, considerând că instanța de mai sus este legală?

# Problema 1 – Soluție

1. Exemple de attribute ce nu reprezinta chei candidat sunt: *nume*, *varsta*. (*nota* nu este un exemplu potrivit luand in considerare instanta data ca exemplu, chiar daca "bunul simt" ne spune ca in general mai multi studenti pot avea aceeasi nota).
2. Nu putem determina o cheie doar pe baza instanței unei relații. Faptul ca instanța este “legală” nu este suficient, o cheie candidat trebuind să fie respectată de toate instanțele legale ale unei relații.

# Problema 2

Fie următoarea structură de bază de date pentru stocarea de informații despre filme

- *Actors* (ID, Name, YoB), unde *ID* este un identificator unic pentru un actor, *Name* și *YoB* fiind numele și anul nasterii actorului;
- *Movies*(ID, Title, Year), unde *ID* este un identificatorul unic al unui film, *Title* și *Year* reprezentând titlul și anul producției filmului;
- *Casting*(MovieID, ActorID, Charac), unde *MovieID* și *ActorID* referă înregistrări din *Movies* și *Actors*. Actorul joacă personajul *Charac* în filmul respectiv.

Exprimați următoarele interogări în SQL:

## Problema 2 – Soluție

1. *Titlurile filmelor produse după 1950 și înainte de 2000;*

```
SELECT Title FROM Movies  
WHERE Year > 1950 AND Year < 2000
```

2. *Titlurile filmelor produse după 2000 sau înainte de 1950*

```
SELECT Title FROM Movies  
WHERE Year < 1950 OR Year > 2000
```



## Problema 2 – Soluție

*3. Numele actorilor ce au jucat personajul "Fletcher Christian" într-o producție a filmului "Mutiny on the Bounty", alături de anul producției filmului;*

```
SELECT Actors.Name, Movies.Year
FROM Actors
      INNER JOIN Casting
            ON Casting.ActorID = Actors.ID
      INNER JOIN Movies
            ON Casting.MovieID = Movies.ID
WHERE Casting.Charac = 'Fletcher Christian'
      AND Movies.Title = 'Mutiny on the Bounty'
```

## Problema 2 – Soluție

4. Găsiți numele actorilor care au jucat personajele “Spiderman” și “Peter Parker” în aceeași producție a unui film;

```
SELECT DISTINCT Actors.Name
FROM Actors
    INNER JOIN Casting as C1
        ON C1.ActorID = Actors.ID
    INNER JOIN Casting as C2
        ON C2.ActorID = Actors.ID
WHERE    C1.Charac = 'Spiderman'
        AND C2.Charac = 'Peter Parker'
        AND C1.MovieID = C2.MovieID
```

## Problema 2 – Soluție

*5. Numele actorilor care au jucat două personaje diferite în același film;*

```
SELECT DISTINCT Actors.Name
FROM Actors
      INNER JOIN Casting as C1
            ON C1.ActorID = Actors.ID
      INNER JOIN Casting as C2
            ON C2.ActorID = Actors.ID
WHERE   C1.Charac <> C2.Charac
      AND C1.MovieID = C2.MovieID
```

## Problema 2 – Soluție

*6. Perechi de nume a doi actori diferiți care au jucat același caracter în producții diferite ale aceleiași film*

```
SELECT A1.Name, A2.Name
FROM Actors A1, Actors A2
INNER JOIN Casting C1 ON C1.ActorID = A1.ID
INNER JOIN Movies M1 ON C1.MovieID = M1.ID
INNER JOIN Casting C2 ON C2.ActorID = A2.ID
INNER JOIN Movies M2 ON C2.MovieID = M2.ID
WHERE M1.Title = M2.Title AND
      M1.ID <> M2.ID AND
      C1.Charac = C2.Charac AND A1.ID<>A2.ID
```

# Problema 3

Fie următoarea structură:

*Suppliers* ( *sid*, *sname*, *address* )

*Products* ( *pid*, *pname*, *color* )

*Catalog* ( *sid*, *pid*, *cost* )

Câmpurile cheie sunt subliniate. Relația *Catalog* conține prețurile practicate de un furnizor pentru un produs particular.

Explicați ce returnează următoarele interogări:

## Problema 3 – Soluție

$$1. \pi_{sname}(\pi_{sid}((\sigma_{color='red'} Products) \otimes (\sigma_{cost < 100} Catalog)) \otimes Suppliers)$$

Numele tuturor furnizorilor ce distribuie produse roșii cu pretul < 100.

$$2. \pi_{sname}(\pi_{sid}((\sigma_{color='red'} Products) \otimes (\sigma_{cost < 100} Catalog) \otimes Suppliers))$$

Interogarea nu returnează nimic, deoarece ultima proiecție nu se poate realiza, câmpul *sname* neexistând (odată ce a fost proiectat *sid* la un pas anterior, acesta este singurul câmp în rezultat).

## Problema 3 – Soluție

$$3. \pi_{sname}((\sigma_{color='red'} Products) \otimes (\sigma_{cost < 100} Catalog) \otimes Suppliers) \cap \\ \pi_{sname}((\sigma_{color='green'} Products) \otimes (\sigma_{cost < 100} Catalog) \otimes Suppliers)$$

Numele tuturor furnizorilor ce distribuie cel puțin un produs roșu cu prețul < 100 și cel puțin un produs verde cu prețul < 100.

$$4. \pi_{sname}((\sigma_{color='red'} Products) \otimes (\sigma_{cost < 100} Catalog) \otimes Suppliers) \cup \\ \pi_{sname}((\sigma_{color='green'} Products) \otimes (\sigma_{cost < 100} Catalog) \otimes Suppliers)$$

Numele tuturor furnizorilor ce distribuie cel puțin un produs roșu sau cel puțin un produs verde la prețul < 100.

## Problema 3 – Soluție

5.  $(\pi_{sid, pid} Catalog) / \pi_{pid} Products$

*(R1 / R2 conține toate tuplurile x a.î. pentru fiecare dintre tuplurile y din R2, există câte un tuplu xy în R1)*

Identificatorii furnizorilor ce distribuie toate produsele.

6.  $\rho(R_1, Catalog), \rho(R_2, Catalog)$   
 $\pi_{R1.pid}(\sigma_{R1.pid = R2.pid \wedge R1.sid \neq R2.sid}(R_1 \times R_2))$

Identificatorii produselor ce sunt distribuite de cel puțin 2 furnizori diferiți



# Problema 4

Pentru înregistrarea cărților dintr-o bibliotecă avem nevoie de următoarele informații:

*bookno, ISBN, section, serial no, sample no, title, author.*

Fiecare exemplar de carte (*sample*) are asociat un număr unic (*bookno*) ce e utilizat pentru înregistrarea împrumuturilor. Acest număr e atașat unei cărți sub forma de cod de bare și nu are o semnificație anume

ISBN este un cod unic asociat unei ediții a unei cărți (aceeași valoare pentru toate exemplarele unei ediții).

Fiecare carte face parte dintr-o secțiune. Într-o secțiune cărți diferite au numere seriale diferite (1, 2, 3... pt fiecare secțiune). Dacă sunt mai multe exemplare ale aceleiași cărți, ele sunt numerotate diferit (folosind *sample no*)

Pentru care dintre următoarele DF putem spune că sunt satisfăcute? Explicați.

# Problema 4 – Soluție

1. *bookno* → ISBN, *section*, *serial no*, *sample no*

**DA:** fiecare carte are un număr unic (*bookno*); *bookno* este cheie

2. ISBN → *bookno*

**NU:** pot exista mai multe exemplare ale aceleași cărți (ele vor avea același ISBN dar valori diferite pentru *bookno*)

3. ISBN → *section*, *serial no*

**DA:** această constrângere păstrează consistența stocării cărților în bibliotecă

# Problema 5

Fie următoarele trei tupluri ale unei instanțe legale a relației  $S$  a cărei structură e formată din 3 attribute:  $A, B, C$  (în această ordine):

$(1,2,3)$

$(4,2,3)$

$(5,3,3)$

1. Despre care dintre următoarele dependențe funcționale putem spune că *nu* sunt respectate de  $S$ ?

a)  $A \rightarrow B$     b)  $BC \rightarrow A$     c)  $B \rightarrow C$

2. Puteți identifica o dependență funcțională ce este satisfăcută de  $S$ ?

# Problema 5 – Soluție

1. *Despre care dintre următoarele dependențe funcționale putem spune ca nu sunt respectate de S?*

a)  $A \rightarrow B$  -

b)  $BC \rightarrow A$  nu e respectată. (1,2,3) & (4,2,3): valori diferite pentru A cu aceleași valori pentru BC

c)  $B \rightarrow C$  -

2. *Puteti identifica o dependență funcțională ce este satisfacuta de S?*

Nu. Pentru a spune că o DF e satisfacută de o relație trebuie să ne asigurăm că aceasta e satisfacută de *toate* instanțele permise (legale) ale relației.

# Problema 6

Fie relația  $R(A, B, C, D, E)$  cu mulțimea de DF

$F = \{AB \rightarrow CDE; AC \rightarrow BDE; B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}.$

1. Găsiți toate cheile lui  $R$ .
2. Determinați o acoperire minimală pentru  $F$ .
3. Este  $R$  în BCNF? Explicați.
4. Determinați o descompunere BCNF a lui  $R$ , cu jonctiuni fara pierderi.
5. Solutia gasită la 4 pastrează dependențele? Explicați.
6. Este  $R$  în 3NF? Explicați.
7. Determinați o descompunere 3NF a lui  $R$  folosind algoritmul descris în curs

# Problema 6 – Soluție

1. *Gasiti toate cheile lui R.*

$F = \{AB \rightarrow CDE; AC \rightarrow BDE; B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$

$A^+ = A$

$B^+ = BCED$

- A nu poate fi dedus din alte attribute  $\Rightarrow$  A apartine cheii

$AB^+ = ABCDE$  - cheie candidat

$AC^+ = ACBDE$  - cheie candidat

$AD^+ = AD,$

$AE^+ = AE$

$ADE^+ = ADE$

# Problema 6 – Soluție

2. Determinați o acoperire minimală pentru  $F$ .

■ O acoperire minimală pentru mulțimea  $F$  de dependente functionale este o mulțime  $G$  de dependente functionale pentru care:

1. Fiecare DF din  $G$  e de forma  $\alpha \rightarrow A$
2. Pt fiecare DF  $\alpha \rightarrow A$  din  $G$ ,  $\alpha$  nu are attribute redundante
3. NU sunt DF redundante în  $G$
4.  $G$  și  $F$  sunt echivalente

Algoritm de calcul al acoperirii minime pt  $F$ :

1. Folosim descomp. pentru a obține DF cu 1 atribut în partea dreapta
2. Se elimină attributele redundante
3. Se elimină dependentele functionale redundante

O acoperire minimală pentru  $F$ :

$\{B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$

# Problema 6 – Soluție

3. Este  $R$  in BCNF? Explicați.

O relație  $R$  ce satisface dependențele funcționale  $F$  se află în *Forma Normală Boyce-Codd* (BCNF) dacă, pentru toate  $\alpha \rightarrow A$  din  $F^+$ :

- $A \in \alpha$  (DF *trivială*), sau
- $\alpha$  conține o cheie a lui  $R$  ( $\alpha$  este o supercheie).

$F = \{AB \rightarrow CDE; AC \rightarrow BDE; B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$

chei candidat:  $AB$  și  $AC$

$R$  nu este in BCNF deoarece exista cel puțin o DF (ex.  $B \rightarrow C$ ) care nu e trivială și partea stângă nu reprezintă o supercheie (nu include o cheie).



# Problema 6 – Soluție

4. *Determinati o descompunere BCNF a lui R, cu jonctiuni fara pierderi.*

Fie relatia R cu DF F. Daca  $\alpha \rightarrow A$  nu respecta definitia BCNF, se descompune R in R - A si  $\alpha A$ .

$F = \{AB \rightarrow CDE; AC \rightarrow BDE; B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$

$B \rightarrow C$  nu respecta BCNF

Decompunem R in  $\{\mathbf{ABDE}\}, \{\mathbf{BC}\}$

$B \rightarrow E$  nu respecta BCNF

Decompunem R in  $\{\mathbf{ABD}\}, \{\mathbf{BC}\}, \{\mathbf{BE}\}$

## Problema 6 – Soluție

5. *Solutia gasita la 4 pastreaza dependentele? Explicati.*

$F = \{AB \rightarrow CDE; AC \rightarrow BDE; B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$

Descompunerea gasita:  $\{ABD\}, \{BC\}, \{BE\}$

Descompunerea relatiei este in BCNF, este cu jonctiuni fara pierderi insa nu pastreaza dependentele (de exemplu  $C \rightarrow D$  nu este respectata deoarece C si D fac parte din relatii distincte.

# Problema 6 – Soluție

6. *Este R in 3NF? Explicati.*

O relație R ce satisface dependențele funcționale F se află în *A Treia Formă Normală (3NF)* dacă, pentru toate  $\alpha \rightarrow A$  din  $F^+$

- $A \in \alpha$  (DF *trivială*), sau
- $\alpha$  este o supercheie pentru R, sau
- A este un atribut prim.

$F = \{AB \rightarrow CDE; AC \rightarrow BDE; B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$

chei candidat: AB si AC

R nu este in 3NF deoarece exista cel putin o DF (ex.  $C \rightarrow D$ ) care nu este triviala, partea stanga nu este o supercheie si  $D \not\subset AB$  sau  $D \not\subset AC$ .

# Problema 6 – Soluție

7. *Determinati o descompunere 3NF a lui R folosind algoritmul descris in curs*

**Input:** Relatia R cu DF  $F$  care este o acoperire minimala

**Output:** O descompunere 3NF a lui R, cu jonctiuni fara pierderi si care pastreaza dependentele

Initializare  $D = \emptyset$

Se aplica *reuniunea* pentru a combina intr-o singura DF toate dependentele din  $F$  ce au acelasi atribut in partea stanga

Pentru fiecare DF  $\alpha \rightarrow \beta$  din  $F$

Se insereaza relatia  $\alpha\beta$  in multimea  $D$

Se insereaza  $\delta$  in  $D$ , unde  $\delta$  este o cheie candidat a lui R

Se elimina relatiile redundante din  $D$  astfel:

sterge  $R_i$  din  $D$  daca  $R_i \subseteq R_j$ , unde  $R_j \in D$

return  $D$

# Problema 6 – Soluție

$\{B \rightarrow C; C \rightarrow B; C \rightarrow D; B \rightarrow E\}$  - acoperire minimala  
AB, AC chei candidat

utilizam reuniunea pentru a combina DS cu aceeasi configuratie in  
partea stanga  $\Rightarrow$   $\{B \rightarrow CE, C \rightarrow BD\}$

Descompunerea lui R:

$\{B;C;E\}$   
 $\{C;B;D\}$   $\rangle$  generate din DF

$\{A;B\}$   
 $\{A;C\}$   $\rangle$  generate din chei

Descompunerea de mai sus este configuratia finala, deoarece nu sunt  
relatii redundante

## Problema 7

Fie un hard disc ce are dimensiunea unui sector egala cu 512 bytes, 2.000 piste pe o față, 50 sectoare per pista, 5 discuri duble, media timpului de cautare de 10 msec.

1. Care este capacitatea unei piste (in bytes)? Care este capacitatea fiecărei suprafețe? Care este capacitatea unui disc?
2. Cati cilindri are harddiscul?
3. 256 bytes e o dimensiune valida a unui bloc? 2.048? 51.200?
4. Daca discurile se rotesc cu 5.400 rpm, care este intarzierea de rotatie maxima?
5. Presupunand ca o pista cu date poate fi transferata printr-o singura miscare de revolutie, care este viteza de transfer?

## Problema 7 – Soluție

*1. Care este capacitatea unei piste (in bytes)? Care este capacitatea fiecărei suprafețe? Care este capacitatea unui disc?*

$$\text{bytes/track} = \text{bytes/sector} * \text{sector/track} = 512 * 50 = 25Kb$$

$$\begin{aligned} \text{bytes/surface} &= \text{bytes/track} * \text{track/surface} = 25 Kb * 2000 = \\ &= 50000 Kb \end{aligned}$$

$$\begin{aligned} \text{bytes/disk} &= \text{bytes/surface} * \text{surfaces} = 50.000Kb * 2 * 5 = \\ &= 500000 Kb \end{aligned}$$

*2. Cati cilindrii are hard disk-ul?*

$$\text{nr. de cilindrii} = \text{nr de piste pe fiecare disc} \rightarrow 2000$$

## Problema 7 – Soluție

3. *256 bytes e o dimensiune valida a unui bloc? 2.048? 51.200?*

Dimensiunea unui bloc este un multiplu al dimensiunii unui sector → doar 2.048 si 51.200 sunt dimensiuni de bloc valide.

4. *Daca discurile se rotesc cu 5.400 rpm, care este intarzierea de rotatie maxima?*

Intarzierea de rotatie maxima = timpul necesar pentru o rotatie  
$$60 / 5400 = 0,011 \text{ sec.}$$

5. *Presupunand ca o pista cu date poate fi transferata printr-o singura miscare de revolutie, care este viteza de transfer?*

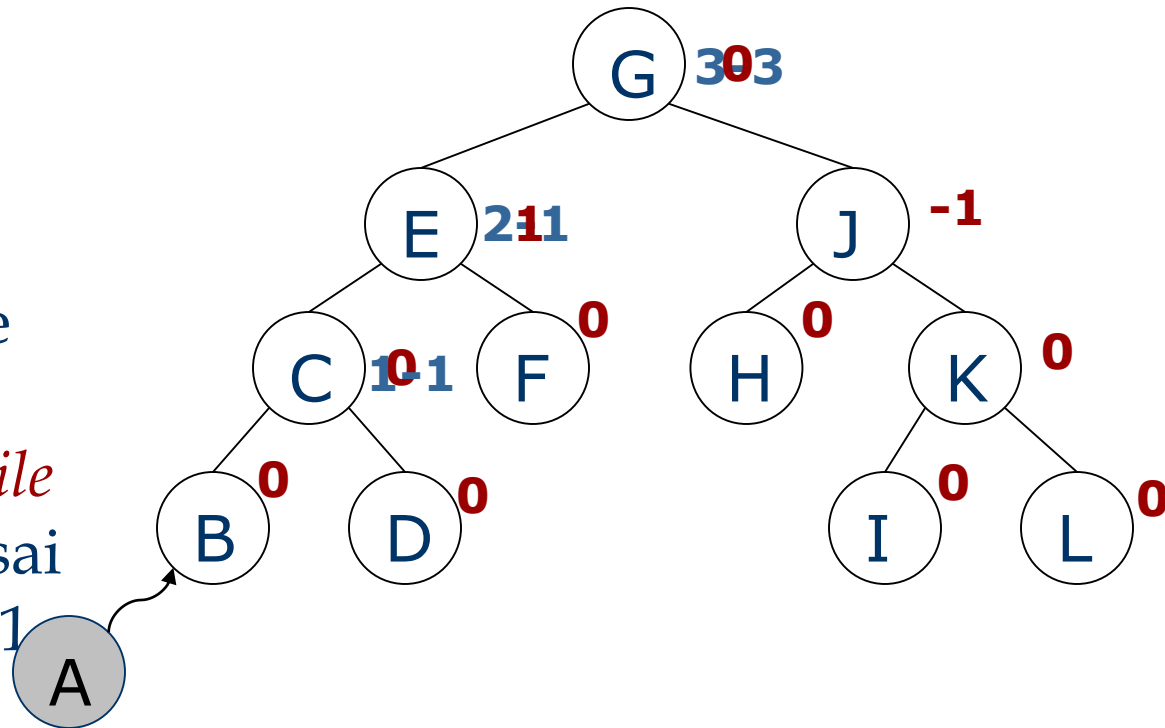
$$25 \text{ Kb} / 0,011 = 2.250 \text{ bytes pe sec}$$



## Problema 8

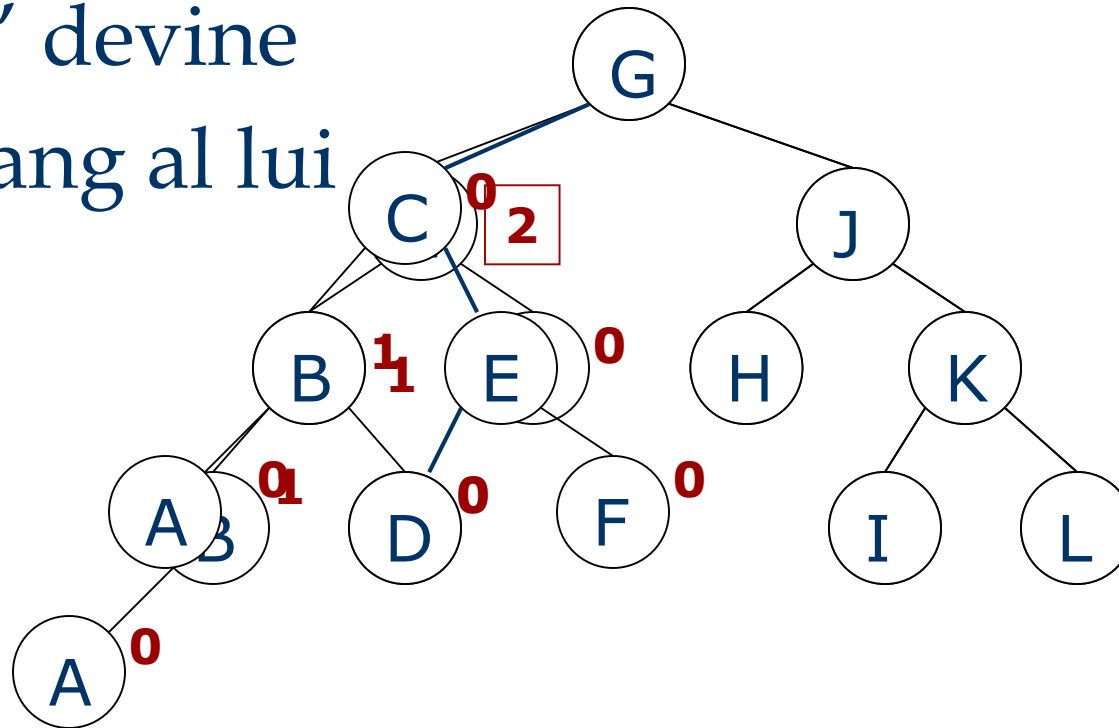
Se considera un arbore binar echilibrat. Descrieti operatiile ce trebuiesc realizate pentru ca arborele sa ramana echilibrat si dupa inserarea valorii 'A'

Arbore  
echilibrat =  
pentru fiecare  
nod diferenta  
dintre *inaltimele*  
subarborilor sai  
este 0, 1 sau -1



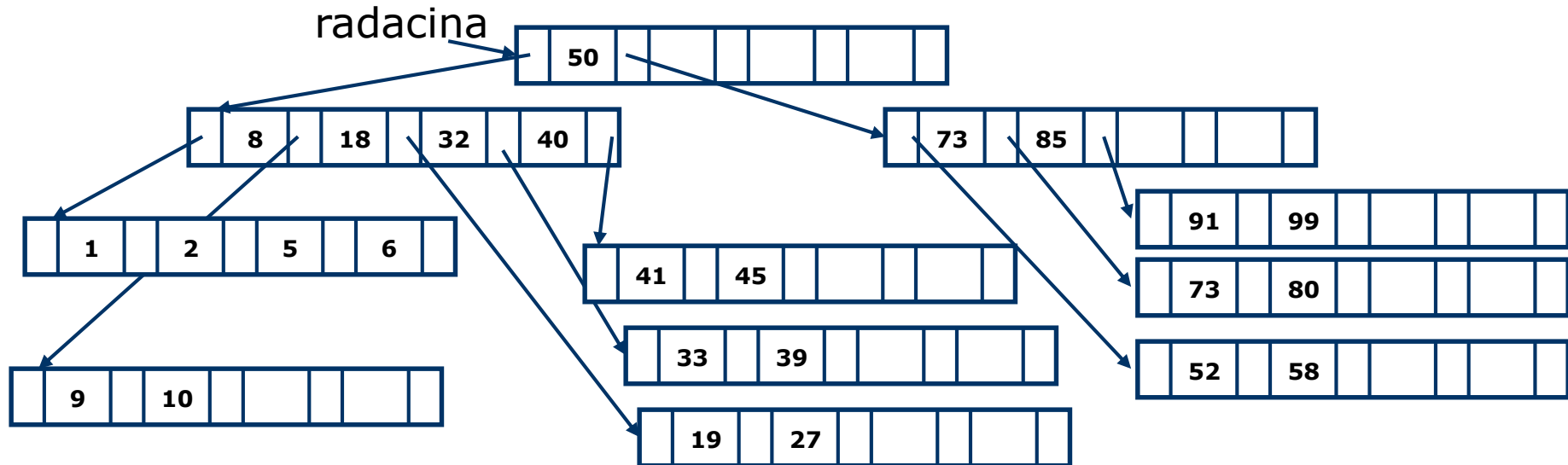
## Problema 8 – Soluție

- nodul 'E' devine copilul drept al lui 'C'
- nodul 'D' devine copilul stang al lui 'E'
- nodul 'C' devine copilul stang al lui 'G'



# Problema 9

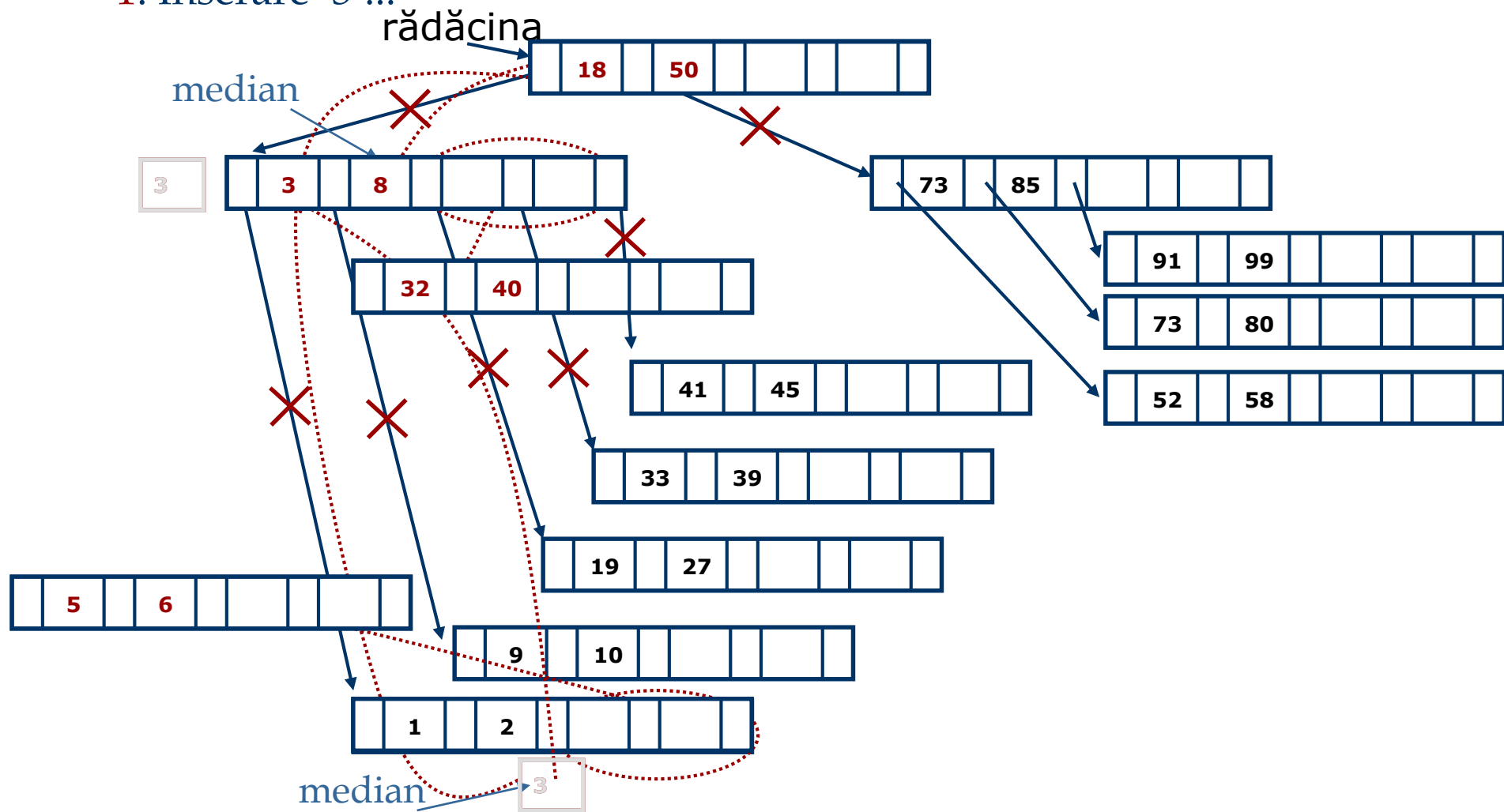
Fie urmatorul arbore - B de ordin 5:



1. Determinați arborele ce rezulta în urma inserării valorii '3'.
2. Determinați arborele ce rezulta în urma eliminării valorii '8' din arborele original (se va folosi redistribuirea).
3. Determinați arborele ce rezulta în urma eliminării valorii '8' din arborele original (se va folosi concatenarea).

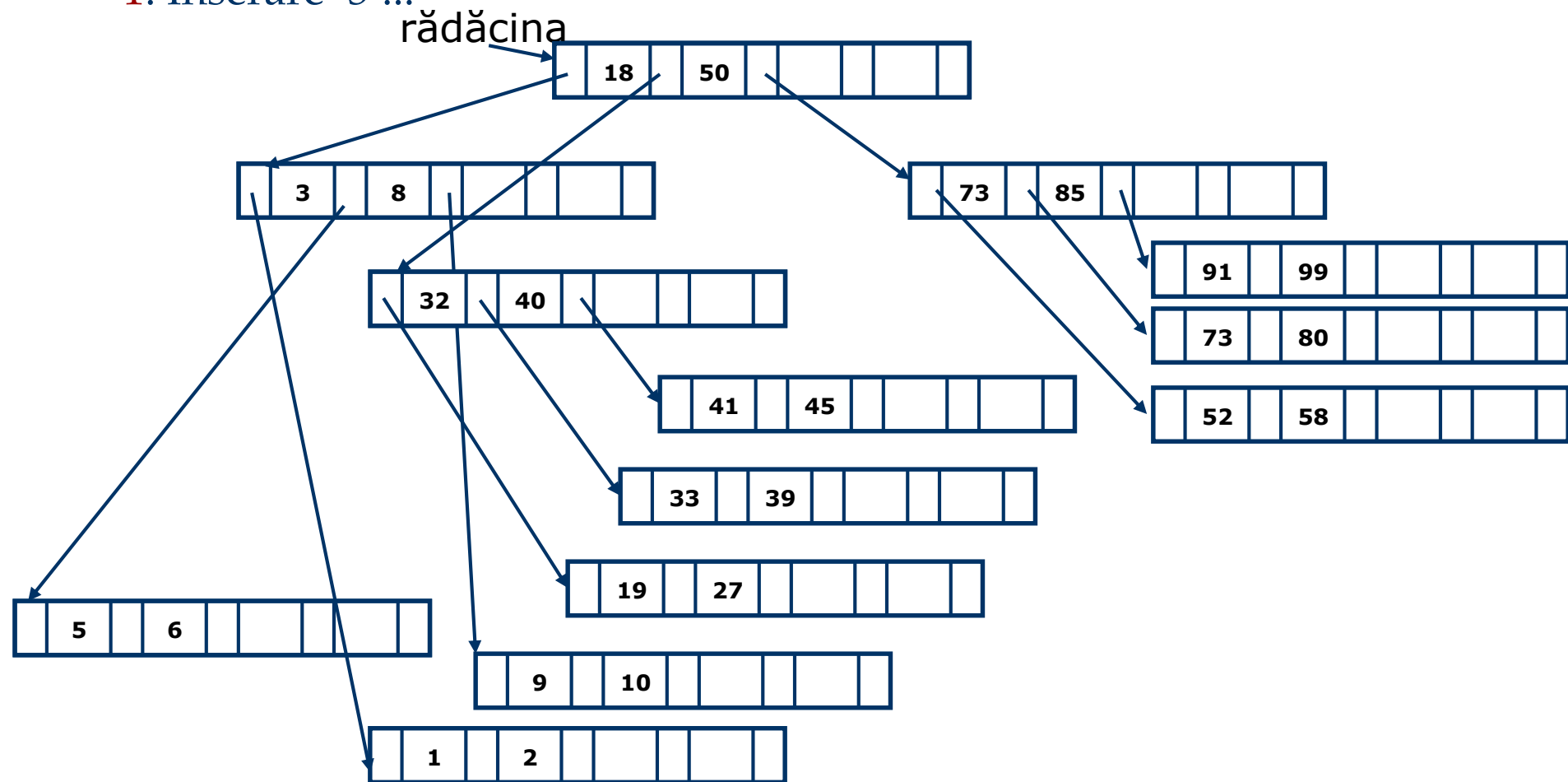
# Problema 9 – Soluție

1. Inserare '3'...



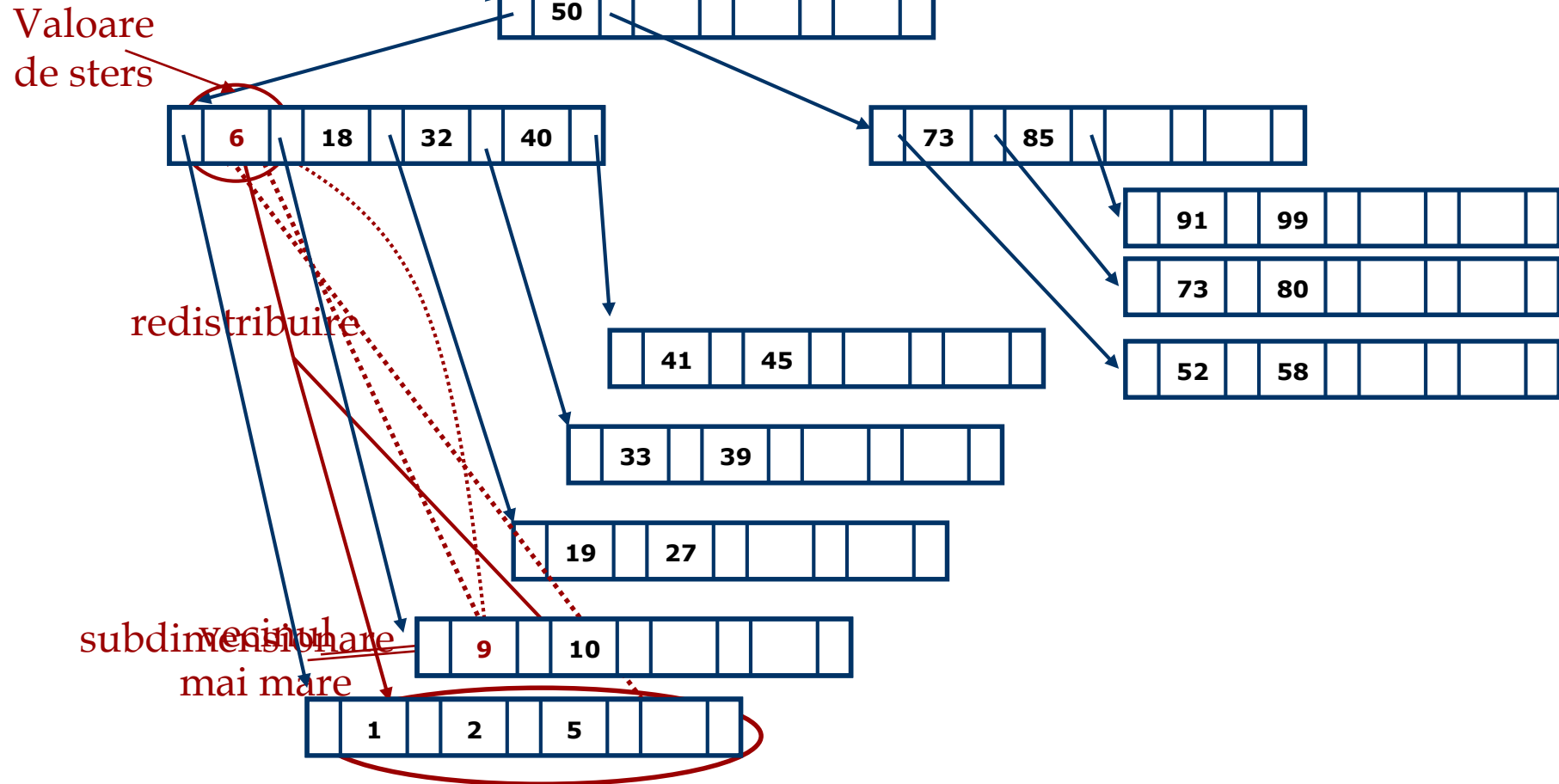
# Problema 9 – Soluție

1. Inserare '3'...



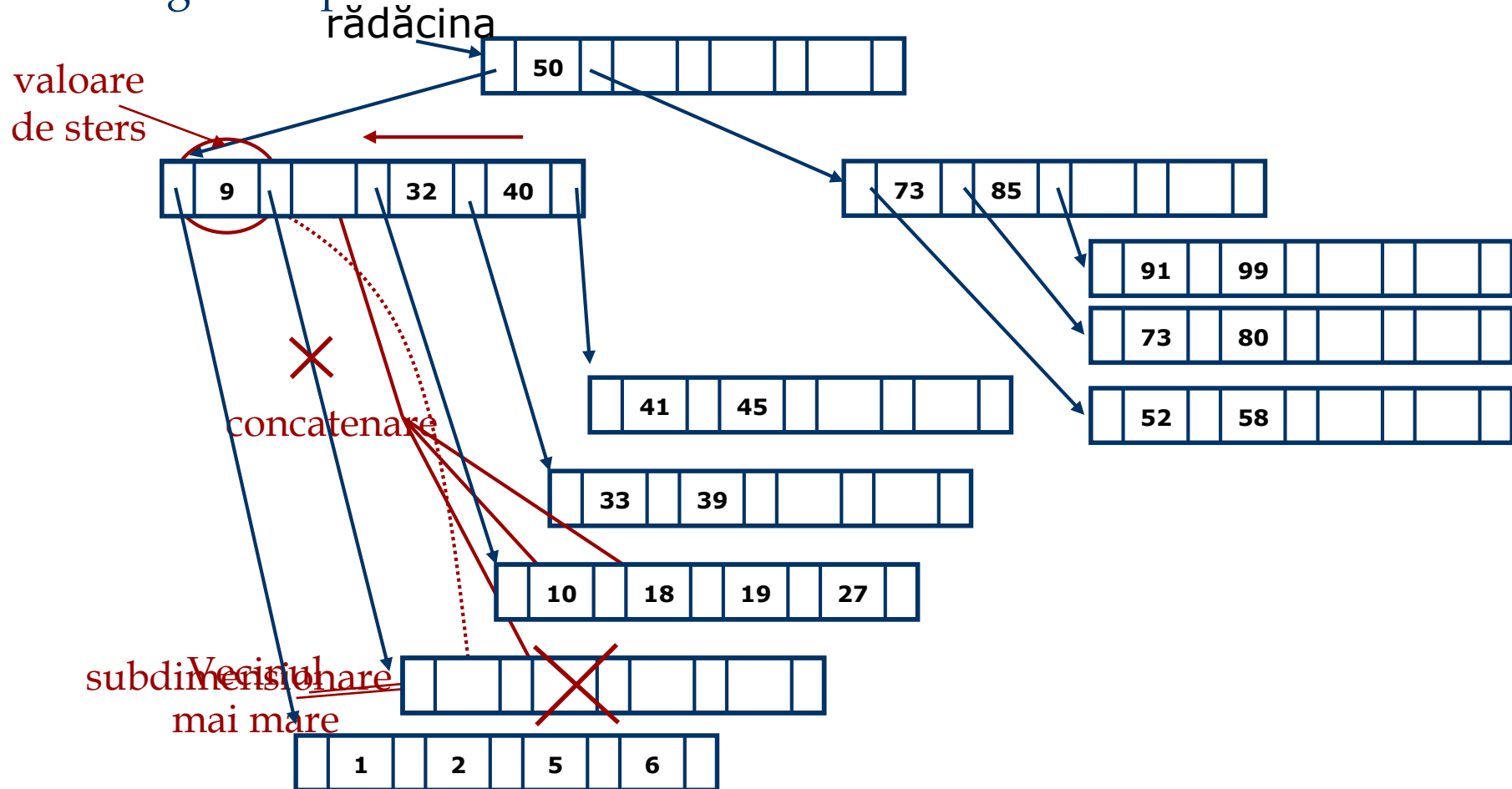
# Problema 9 – Soluție

## 2. Stergere '8' prin redistribuire rădăcina



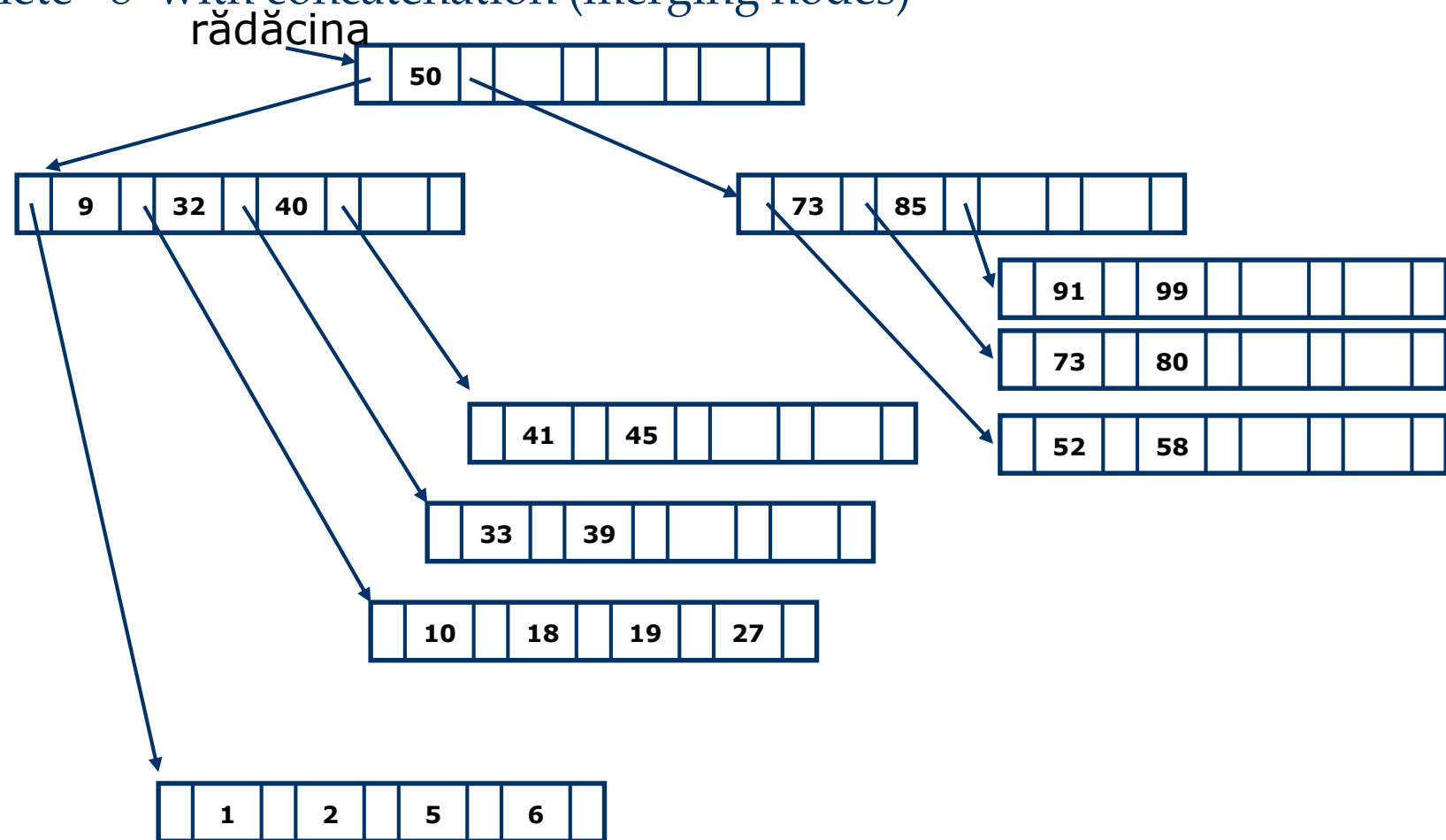
# Problema 9 – Soluție

## 3. Stergere '8' prin concatenare



# Problema 9 – Soluție (cont)

3. Delete '8' with concatenation (merging nodes)





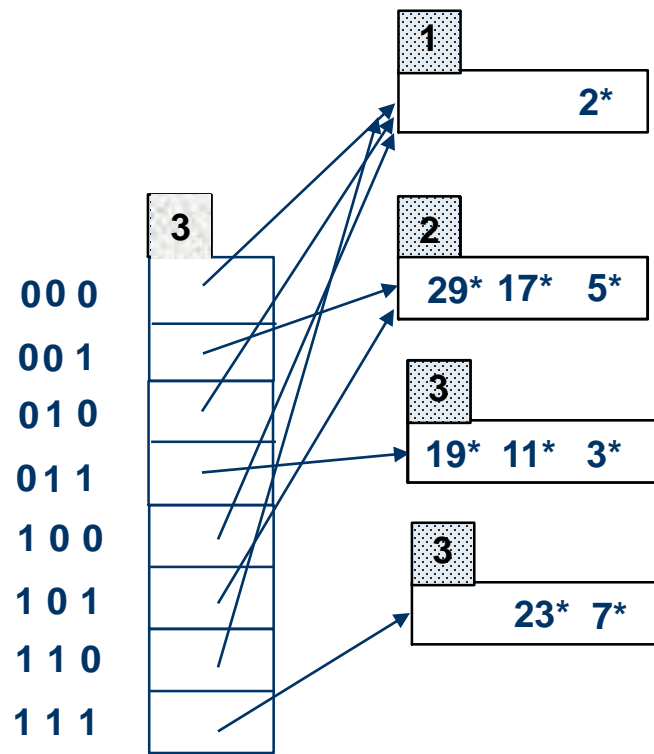
## Problema 10

Se utilizeaza un fisier cu organizare directă extensibil (*extendible hashing*) pentru o tabelă cu următoarele valori pentru cheia de căutare:

2, 3, 5, 7, 11, 17, 19, 23, 29.

Determinați structura fișierului dacă funcția de dispersie utilizată este  $h(x) = x \bmod 8$  și fiecare pagină de memorie poate conține maxim 3 valori.

# Problema 10 – Soluție



x	x mod 8	bin
2	2	010
3	3	011
5	5	101
7	7	111
11	3	011
17	1	001
19	3	011
23	7	111
29	5	101