Modelul Relaţional

Teoria relațională

Algebra Relaţională – partea 1

Modelul Relaţional

- E.F. Codd, 1970 (1972)
- System R, 1977 (IBM)
- Oracle, 1979

Modelul Relaţional

- **<u>Relaţie</u>**: Fiind dată o colecţie de mulţimi D_1, D_2, … D_n (nu neapărat distincte), se spune că R este o relaţie pe aceste mulţimi dacă este o mulţime de n-tuple (d_1, d_2, … d_n) astfel încât d_i aparţine D_i, i=1..n
- Mulţimile D_1, D_2, … D_n sunt domeniile relaţiei R.
- n este **gradul** sau **aritatea** relaţiei R.
- Numărul de n-tuple reprezintă **cardinalitatea** relaţiei R.

Modelul Relaţional

- **<u>Relaţie</u>** (a doua definiţie): Se defineşte produsul cartezian D_1x D_2x …x D_n al mulţimilor D_1, D_2, … D_n mulţimea tuturor n-tuplelor ordonate (d_1, d_2, … d_n) astfel încât d_1 aparţine D_1, d_2 aparţine D_2, …, d_n aparţine D_n.

  O relaţie R pe mulţimile D_1, D_2, … D_n este o submulţime a produsului cartezian D_1x D_2x …x D_n.

Modelul Relaţional - Paradigme

Relaţia este o mulţime (set) de n-tuple:

1. Nu există două elemente (n-tuple) identice.
2. Ordinea elementelor este indiferentă.

Modelul Relaţional

- **<u>Domeniu</u>**: Ansamblul de valori admisibile pentru o componentă a unei relaţii.
- Exemple:
  - Domeniul numelor de persoane
  - Domeniul numelor de oraşe
  - Domeniul notelor (mulţimea {1, 2, 3, 4, 5, 6, 7, 8, 9, 10})
- Domenii compatibile: Mulţimile de valori care le definesc sunt comparabile d.p.d.v. semantic.
- **<u>Atribut</u>**: Un domeniu cu nume, adică utilizarea unui domeniu sub un nume oarecare (într-o relaţie).

Modelul Relaţional

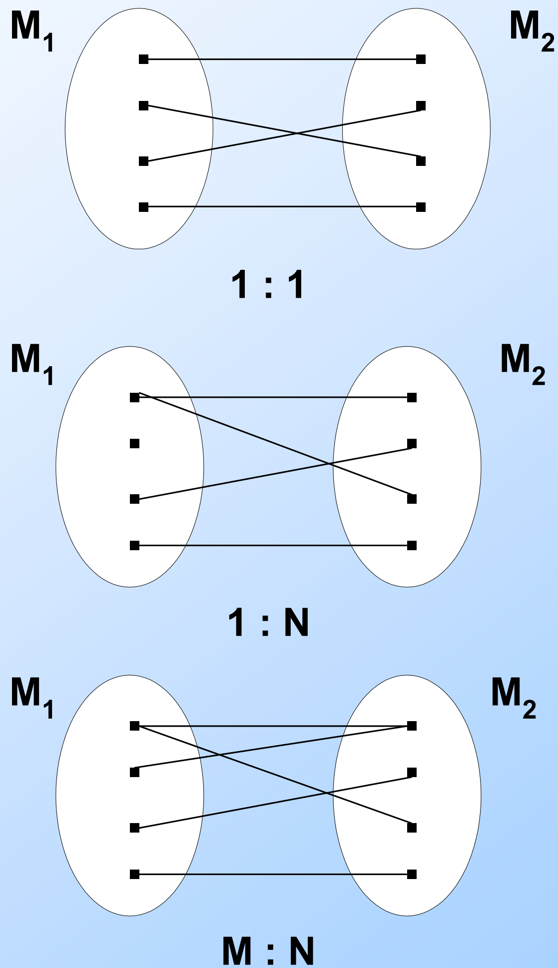
<u>Cheie</u>: Se numeşte cheie a unei relaţii R, un subset K al atributelor relaţiei R ce satisface proprietăţile:

1. Identificare unică, fiecare tuplă a relaţiei R este identificată în mod unic de valorile atributelor care compun cheia K
2. Neredundanţă, subsetul K este minimal în sensul că eliminarea oricărui atribut din K duce la pierderea proprietăţii 1.

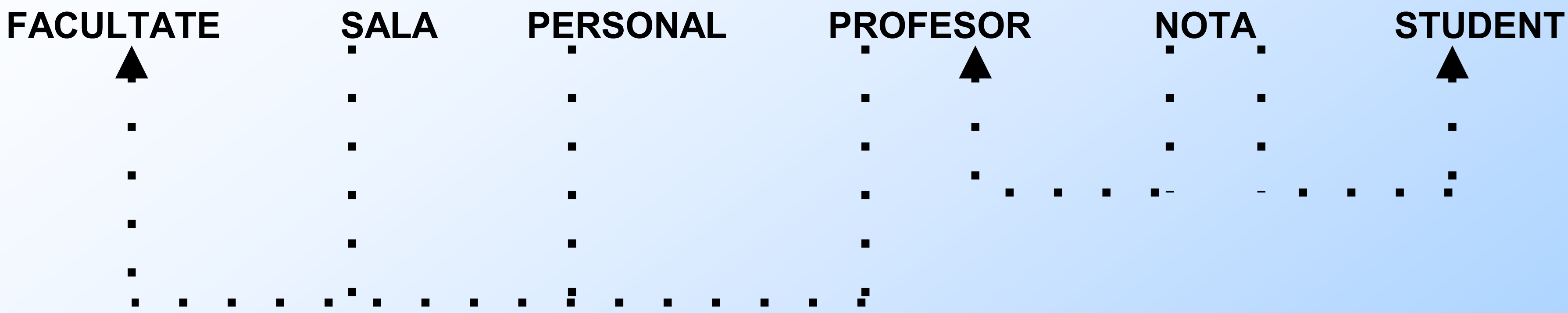
Modelul Relaţional

- Tipuri de chei:
  - Primară
  - Candidată
  - Străină
- **Atribut prim**: Atribut constituent al unei chei.
- O cheie poate fi *simplă* când are un singur atribut sau *compusă* când este formată din mai multe atribute.

Tipuri de legături



Arborele de structură a datelor (modelul relaţional)



Schema BD relaţionale cuprinde relaţii şi legături relaţionale realizate prin ***valoare*** (chei străine) şi NU prin pointeri ca în modelele ierarhic sau reţea, de aceea structura de date este de tip NIVEL, toate nodurile sunt pe acelaşi nivel, fiecare nod reprezintă o relaţie. Legăturile M:N se implementează prin introducerea unei RELAŢII DE LEGĂTURĂ (NOTA).

SGBD total relaţional

- Principiul integrităţii domeniului
- Principiul integrităţii relaţiei
- Principiul integrităţii referinţei
- LMD cel puţin echivalent cu algebra relaţională

“Algebră”

- Sistem matematic ce constă din:
  - *Operanzi* --- variabile sau valori din care se construiesc valori noi.
  - *Operatori* --- simboluri ce denotă procedurile ce construiesc valorile noi din valori existente.

Algebra Relaţională

- Este o algebră ai cărei operanzi sunt relaţii sau variabile ce reprezintă relaţii.
- Operatorii sunt concepuţi astfel încât să fie efectuate operaţiile dorite cu relaţiile din BD.
  - Rezultatul este o algebră ce poate fi utilizată ca un *limbaj de interogare* pentru relaţii.

Esenţa Algebrei Relaţionale

- Uniune, intersecţie şi diferenţă.
  - Operaţiile obişnuite pe mulţimi, dar *ambii operanzi trebuie să aibă aceeaşi schemă de relaţie*.
- Selecţie: alege anumite rânduri.
- Proiecţie: alege anumite coloane.
- Produs şi join: compun din relaţii.
- Redenumire relaţii şi atribute.

Operatori Primitive (cinci)

- Reuniunea
- Diferența
- Produsul cartezian
- Selecția
- Proiecția

Selecţie

- $R1 := \sigma_C(R2)$
  - *C* este o condiţie (asemănător cu instrucţiunea "if") ce face referire la atributele din R2.
  - R1 conţine acele tuple din R2 ce satisfac *C*.

Exemplu: Selecţie

Relaţia Sells:

bar	beer	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Miller	3.00

JoeMenu := $\sigma_{bar=\text{“Joe's”}}$(Sells):

bar	beer	price
Joe's	Bud	2.50
Joe's	Miller	2.75

Proiecţie

- R1 := π_L (R2)
  - L este o listă de atribute din schema relaţiei R2.
  - R1 este construită în felul următor:
    - Mai întâi pentru fiecare tuplă din R2 se extrag atributele din lista L, în ordinea specificată.
    - Se elimină tuplele duplicat, dacă există.

Exemplu: Proiecţie

Relaţia Sells:

bar	beer	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Miller	3.00

Prices := $\pi_{beer,price}$(Sells):

beer	price
Bud	2.50
Miller	2.75
Miller	3.00

Proiecţie Extinsă

- Se foloseşte acelaşi operator π_L, dar se permit în lista L expresii arbitrare ce implică atribute:
  1. Expresii aritmetice cu atribute, de exemplu: $A+B\text{->}C$.
  2. Duplicarea atributelor (un atribut să apară de mai multe ori).

Exemplu: Proiecţie Extinsă

R = (

A	B
1	2
3	4

)

$\pi_{A+B\text{->}C,A,A}$ (R) =

C	A1	A2
3	1	1
7	3	3

Produs

- R3 := R1 X R2
  - Face pereche între fiecare tuplă t1 din R1 şi fiecare tuplă t2 din R2.
  - O tuplă din R3 se obţine prin concatenarea t1t2.
  - Schema relaţiei R3 este constituită din atributele din R1 şi apoi din R2, în ordine.
  - Atenţie la atributele cu acelaşi nume în R1 şi R2, de exemplu *A* : se foloseşte exprimarea R1.*A* şi R2.*A*.

Exemplu: R3 := R1 X R2

R1(

A,	B)
1	2
3	4

R2(

B,	C)
5	6
7	8
9	10

R3(

A,	R1.B,	R2.B,	C)
1	2	5	6
1	2	7	8
1	2	9	10
3	4	5	6
3	4	7	8
3	4	9	10

Theta-Join

- R3 := R1 $\bowtie_C$ R2
  - Se face produsul R1 X R2.
  - Apoi se aplică σ_C rezultatului.
- Ca şi pentru σ, C poate fi orice condiţie cu valoare booleană.
  - Versiuni istorice ale acestui operator permiteau doar A θ B, unde θ este =, <, etc.; de unde şi numele "theta-join."

Exemplu: Theta Join

Sells(

bar,	beer,	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

)

Bars(

name,	addr
Joe's	Maple St.
Sue's	River Rd.

)

BarInfo := Sells $\bowtie_{\text{Sells.bar = Bars.name}}$ Bars

BarInfo(

bar,	beer,	price,	name,	addr
Joe's	Bud	2.50	Joe's	Maple St.
Joe's	Miller	2.75	Joe's	Maple St.
Sue's	Bud	2.50	Sue's	River Rd.
Sue's	Coors	3.00	Sue's	River Rd.

)

Natural Join

- Este o variantă folositoare (join *natural*) conectează două relaţii prin:
  - Egalizarea atributelor cu acelaşi nume şi
  - Proiecţia unei singure copii a fiecărui atribut pereche (unul din atributele egalizate).
- Notaţie R3 := R1 $\bowtie$ R2.

Exemplu: Natural Join

Sells(

bar,	beer,	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

)

Bars(

bar,	addr
Joe's	Maple St.
Sue's	River Rd.

)

BarInfo := Sells $\bowtie$ Bars

Notă: Bars.name a devenit Bars.bar pentru a face posibil natural join.

BarInfo(

bar,	beer,	price,	addr
Joe's	Bud	2.50	Maple St.
Joe's	Milller	2.75	Maple St.
Sue's	Bud	2.50	River Rd.
Sue's	Coors	3.00	River Rd.

)

Redenumire

- Operatorul ρ redefineşte schema unei relaţii.
- R1 := $\rho_{R1(A1,\ldots,An)}$(R2) produce R1, o relaţie cu atributele A1,…,A*n* şi aceleaşi tuple ca şi R2.
- Notaţia simplificată : R1(A1,…,A*n*) := R2.

Exemplu: Redenumire

Bars(

name,	addr
Joe's	Maple St.
Sue's	River Rd.

)

R(bar, addr) := Bars

R(

bar,	addr
Joe's	Maple St.
Sue's	River Rd.

)

Construirea de Expresii Complexe

- Se combină operatorii folosind paranteze şi reguli de precedenţă.
- Există trei notaţii, ca şi la expresiile aritmetice:
  1. Secvenţe de instrucţiuni de atribuire.
  2. Expresii cu mai mulţi operatori.
  3. Arbori expresie.

Secvenţe de Atribuiri

- Sunt create nume de relaţii temporare.
- Redenumirea poate fi implicată de acordarea relaţiilor a unei liste de atribute.
- Exemplu: R3 := R1 $\bowtie_C$ R2 poate fi rescrisă:

  R4 := R1 X R2

  R3 := σ_C(R4)

Expresii într-o Singură Atribuire

- Exemplu: operaţia theta-join R3 := R1 $\bowtie_C$ R2 poate fi rescrisă: R3 := σ_C (R1 X R2)
- Precedenţa operatorilor relaţionali:
  1. [σ, π, ρ] (cei mai prioritari).
  2. [x, ⋈].
  3. ∩.
  4. [∪, —]

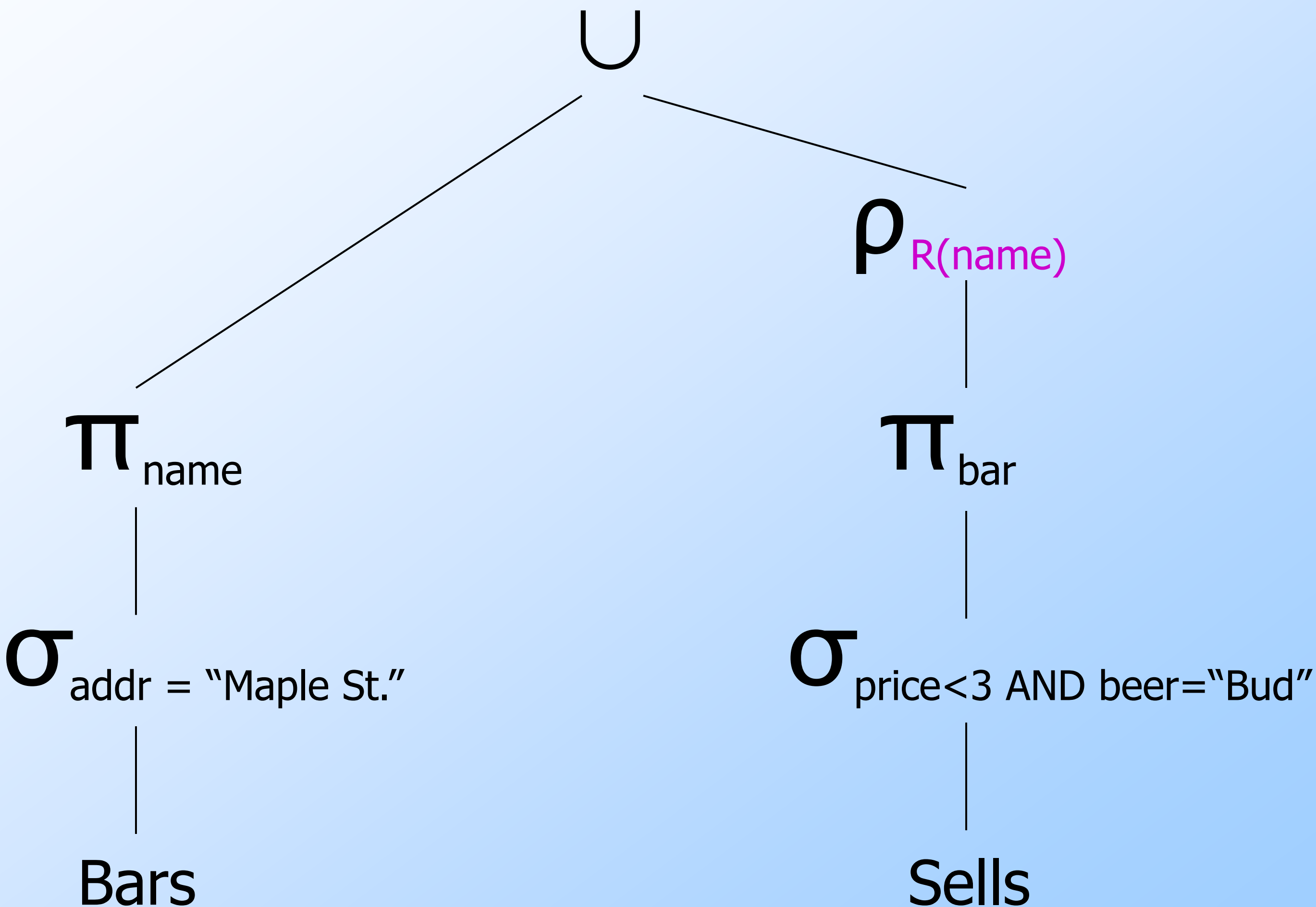
Arbori Expresie

- Frunzele sunt operanzi:
  - variabile ce precizează relaţii;
  - în particular, relaţii constante.
- Nodurile interioare sunt operatori, aplicaţi nodurilor fiu.

Exemplu: Arborele unei Interogări

- Se folosesc relaţiile Bars(name, addr) şi Sells(bar, beer, price), pentru a găsi numele barurilor ce fie se găsesc pe "Maple St." fie vând "Bud" mai ieftin de 3 ($).

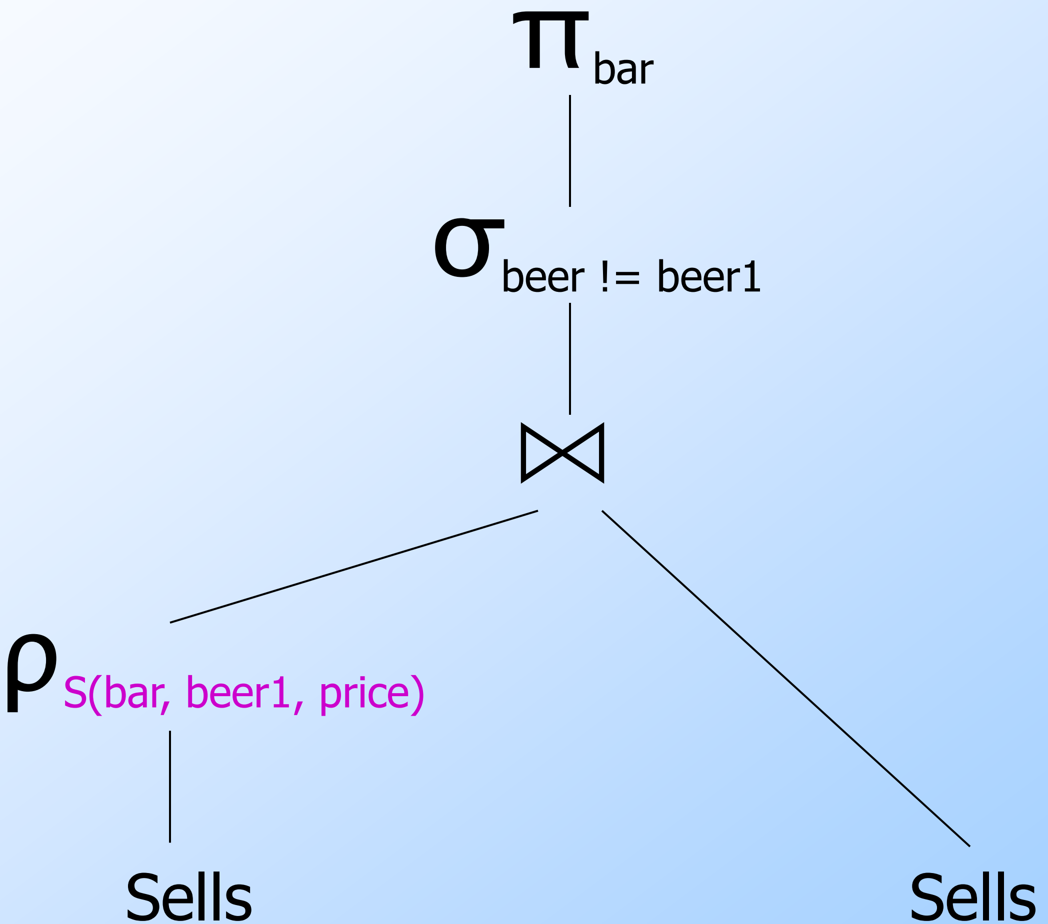
Arborele:


∪
ρ R(name)
π name
π bar
σ addr = “Maple St.”
σ price<3 AND beer=“Bud”
Bars
Sells

Exemplu: Self-Join

- Se foloseşte Sells(bar, beer, price), pentru a găsi barurile ce vând două beri diferite la acelaşi preţ.
- Strategia: prin redenumire, se defineşte o copie a relaţiei Sells, numită S(bar, beer1, price). Join natural între Sells şi S constă din cvadruplele (bar, beer, beer1, price) astfel încât barul vinde ambele beri (“beer” şi “beer1”) la preţul “price”.

Arborele



π_{bar}
$\sigma_{beer\ !=\ beer1}$
$\bowtie$
$\rho_{S(bar,\ beer1,\ price)}$
Sells
Sells

Scheme pentru Rezultate

- Uniune, intersecţie şi diferenţă: schemele celor doi operanzi trebuie să fie aceleaşi, aşa că se foloseşte acea schemă pentru rezultat.
- Selecţie: schema rezultatului este aceeaşi cu schema operandului.
- Proiecţie: lista atributelor dă schema.

Scheme pentru Rezultate

- Produs: schema este constituită din atributele ambelor relaţii.
  - Se foloseşte R.*A*, etc., pentru a distinge două atribute numite *A*.
- Theta-join: asemănător cu produs.
- Natural join: uniunea atributelor celor două relaţii.
- Redenumire: operatorul dă schema.

Algebra Relaţională pe “Bags”

- Un *bag* (sau *multiset*) este asemănător unui set, dar un element poate apare de mai multe ori.
- Exemplu: {1,2,1,3} este un bag.
- Exemplu: {1,2,3} este un bag ce se întâmplă să fie şi un set.

De ce “Bags”?

- SQL, limbajul de interogare cel mai important pentru BD relaţionale, este de fapt un limbaj pe “bag”.
- Anumite operaţii, cum este proiecţia, sunt mult mai eficiente pe bag-uri decât set-uri.

Operaţii pe Bag-uri

- Selecţia se aplică fiecărei tuple, astfel încât efectul pe bag-uri este acelaşi cu efectul pe set-uri.
- Proiecţia se aplică de asemenea fiecărei tuple, dar ca operator "bag", nu se elimină duplicatele.
- Produsul şi join-urile sunt efectuate pe fiecare pereche de tuple, deci duplicatele din bag-uri nu au efect asupra modului de operare.

Exemplu: Selecţie "Bag"

R(

A,	B
1	2
5	6
1	2

)

$\sigma_{A+B<5}(R) =$

A	B
1	2
1	2

Exemplu: Proiecţie "Bag"

R(A, B)

A,	B
1	2
5	6
1	2

$\pi_A(R) =$

A
1
5
1

Exemplu: Produs “Bag”

R(

A,	B
1	2
5	6
1	2

)

S(

B,	C
3	4
7	8

)

R X S =

A	R.B	S.B	C
1	2	3	4
1	2	7	8
5	6	3	4
5	6	7	8
1	2	3	4
1	2	7	8

Exemplu: Theta-Join "Bag"

R(

A,	B
1	2
5	6
1	2

)

S(

B,	C
3	4
7	8

)

$R \bowtie_{R.B<S.B} S =$

A	R.B	S.B	C
1	2	3	4
1	2	7	8
5	6	7	8
1	2	3	4
1	2	7	8

Uniune “Bag”

- Un element apare în uniunea a două bag-uri suma numărului de apariţii din fiecare bag.
- Exemplu: $\{1,2,1\} \cup \{1,1,2,3,1\} = \{1,1,1,1,1,2,2,3\}$

Intersecţie “Bag”

- Un element apare în intersecţia a două bag-uri de minimul numărului de apariţii în cele două bag-uri.
- Exemplu: $\{1,2,1,1\} \cap \{1,2,1,3\} = \{1,1,2\}$.

Diferenţa “Bag”

- Un element apare în diferenţa $A - B$ a două bag-uri de atâtea ori cât apare în A, minus numărul de apariţii în B.
  - Dar niciodată mai puţin de 0 ori.
- Exemplu: {1,2,1,1} – {1,2,3} = {1,1}.

Atenţie: Legile Bag!= Legile Set

- Anumite reguli, dar *nu toate* regulile algebrice ce sunt valabile pentru set-uri sunt de asemenea valabile pentru bag-uri.
- Exemplu: comutativitatea uniunii ($R \cup S = S \cup R$) *se păstrează* pentru bag-uri.
  - Deoarece adunarea este comutativă, prin adăugarea numărului de apariţii ale lui x în R şi S nu depinde de ordinea lui R şi S.

Exemplu: Regulă ce nu se păstrează

- Uniunea set-urilor este *idempotentă*, ceea ce înseamnă că $S \cup S = S$.
- Pentru bag-uri, dacă x apare de n ori în S, atunci el apare de $2n$ ori în $S \cup S$.
- Astfel $S \cup S$!= S în general.
  - adică, {1} ∪ {1} = {1,1} != {1}.