

## Tutoriat 12

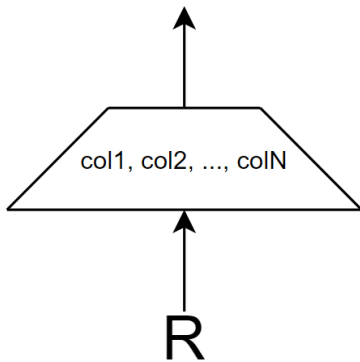
### ARBORELE ALGEBRIC AL UNEI CERERI;

#### ARBORELE ALGEBRIC AL UNEI CERERI

O expresie algebrica mai poate fi exprimata si grafic, sub forma unui arbore unde fiecare operator este reprezentat de un simbol.

#### Operatorul PROJECT

Simbol:



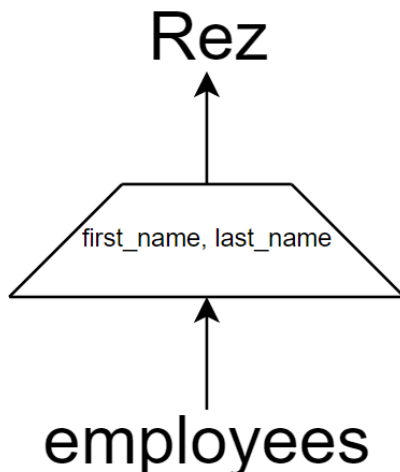
Unde R este fie un tabel, fie o relatie rezultata dintr-un alt operator.

In continuare sa vedem un exemplu prin care vrem sa obtinem first\_name si last\_name din tabelul **EMPLOYEES**.

Expresia algebrica:

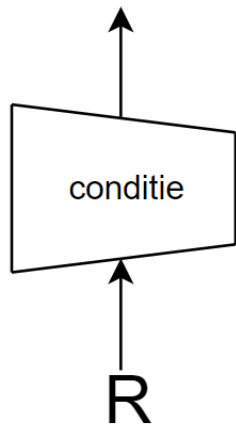
Rez = **PROJECT**(employees, first\_name, last\_name)

Arborele algebric:



## Operatorul SELECT

Simbol:



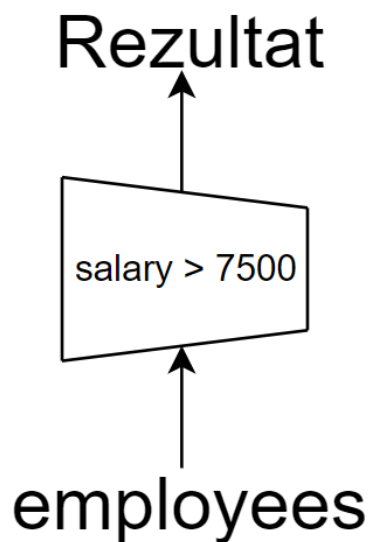
Unde R este fie un tabel, fie o relatie rezultata dintr-un alt operator.

Sa exemplificam acest operator afisand toti angajatii din **EMPLOYEES** ce au un salariu mai mare de 7500.

Expresie algebrica:

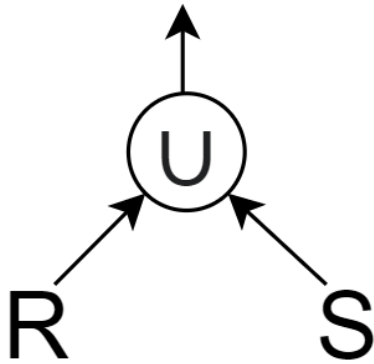
Rezultat = **SELECT**(employees, salary > 7500)

Arbore algebric:



## Operatorul UNION

Simbol:



Unde R si S sunt fie tabele, fie relatii rezultate dintr-un alt operator.

Sa exemplificam acest operator afisand toti angajatii ce au prenumele „John” sau „Steve”.

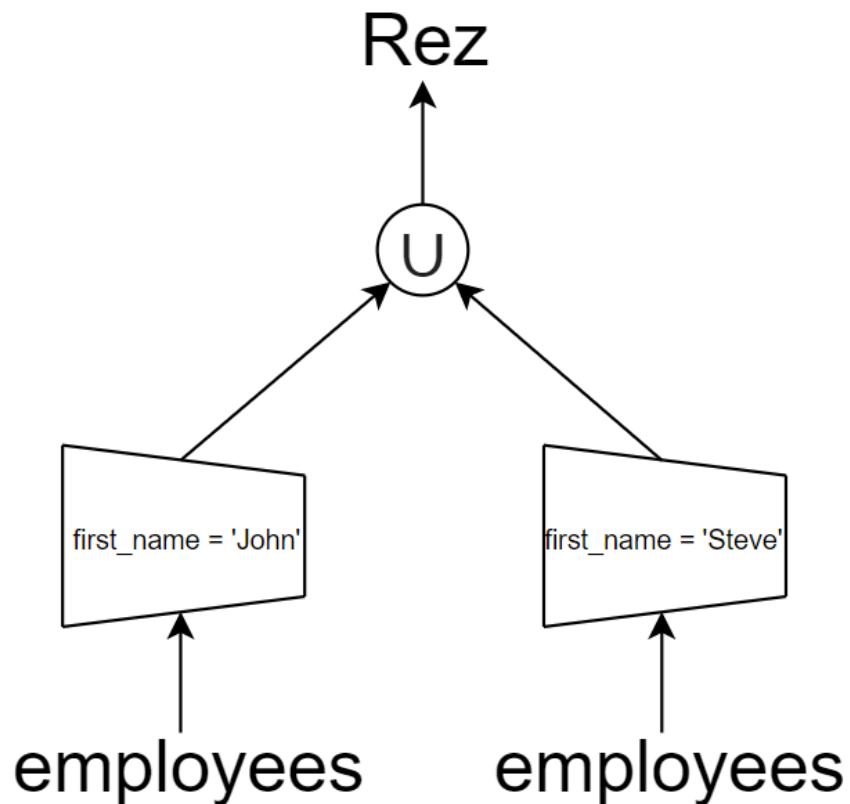
Expresie algebrica:

$R1 = \text{SELECT}(\text{employees}, \text{first\_name} = \text{„John”})$

$R2 = \text{SELECT}(\text{employees}, \text{first\_name} = \text{„Steve”})$

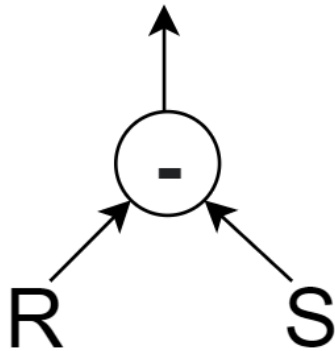
$\text{Rez} = \text{UNION}(R1, R2)$

Arbore algebric:



## Operatorul DIFFERENCE

Simbol:



Unde R si S sunt fie tabele, fie relatii rezultate dintr-un alt operator.

Pentru acest operator ordinea conteaza.

Ca exemplu sa afisam toti angajatii cu employee\_id par ce primesc comision

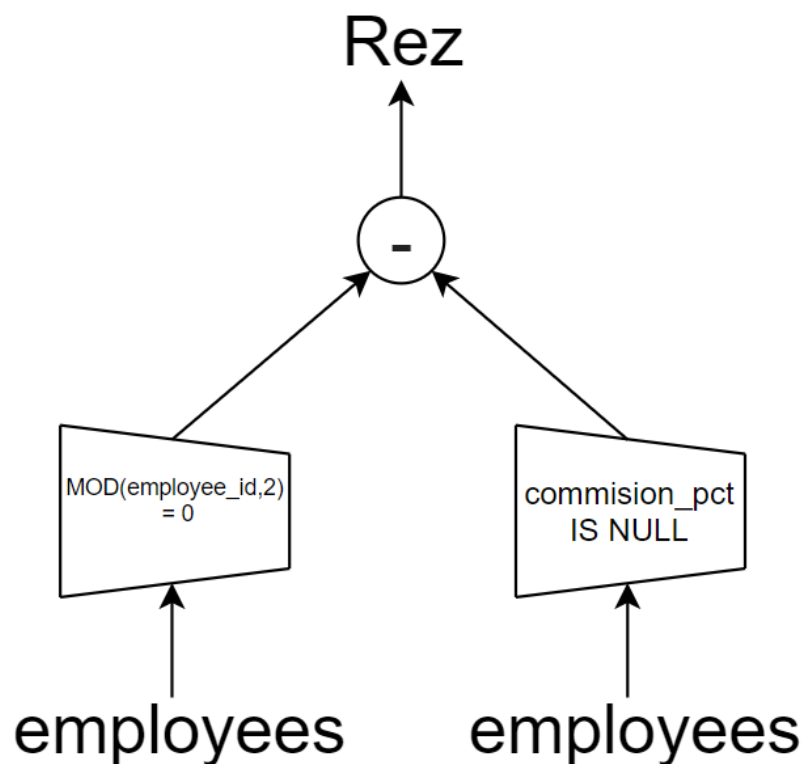
Expresie algebrica:

R1 = `SELECT`(employees, MOD(employee\_id , 2) = 0)

R2 = `SELECT`(employees, commission\_pct IS NULL)

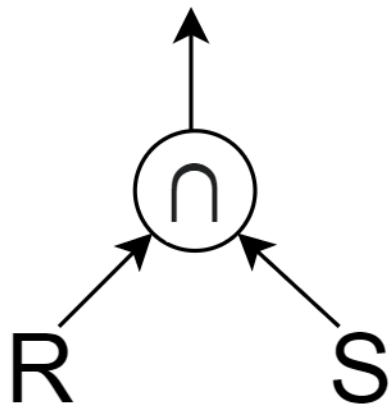
Rez = `DIFFERENCE`(R1,R2)

Arbore algebric:



## Operatorul INTERSECT

Simbol:



Unde R si S sunt fie tabele, fie relatii rezultate dintr-un alt operator.

Ca exemplu sa afisam toti angajatii ce lucreaza ca si „IT\_PROG” in departamentul 60.

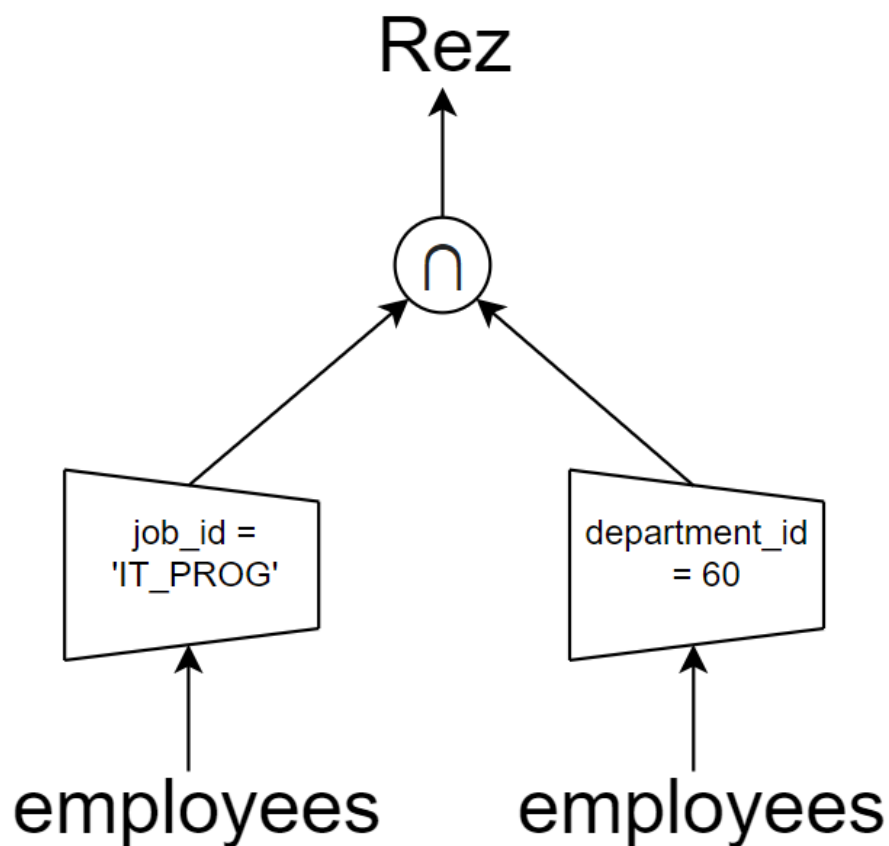
Expresie algebrica:

$R1 = \text{SELECT}(\text{employees}, \text{job\_id} = \text{„IT\_PROG”})$

$R2 = \text{SELECT}(\text{employees}, \text{department\_id} = 60)$

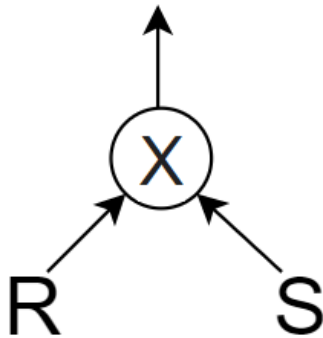
$\text{Rez} = \text{INTERSECT}(R1, R2)$

Arbore algebric:



## Operatorul PRODUCT

Simbol:



Unde R si S sunt fie tabele, fie relatii rezultate dintr-un alt operator.

Ca si exemplu sa afisam pentru fiecare angajat toate departamentele in care poate lucra. Sa se afisaze pentru angajat first\_name si last\_name si pentru departamente doar numele acestora.

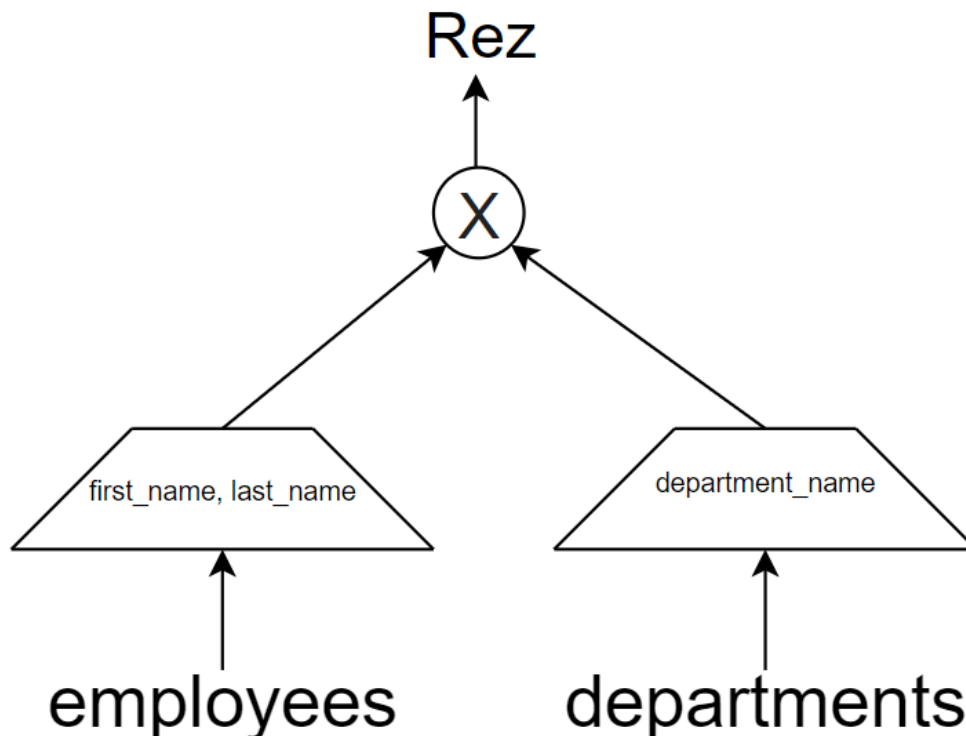
**Expresie algebrica:**

$R1 = \text{PROJECT}(\text{employees}, \text{first\_name}, \text{last\_name})$

$R2 = \text{PROJECT}(\text{departments}, \text{department\_name})$

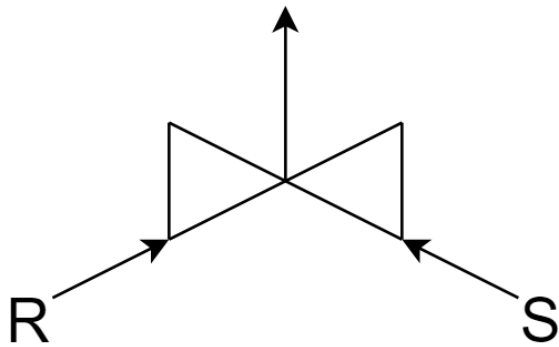
$\text{Rez} = \text{PRODUCT}(R1, R2)$

**Arbore algebric:**



## Operatorul NATURAL JOIN

Simbol:



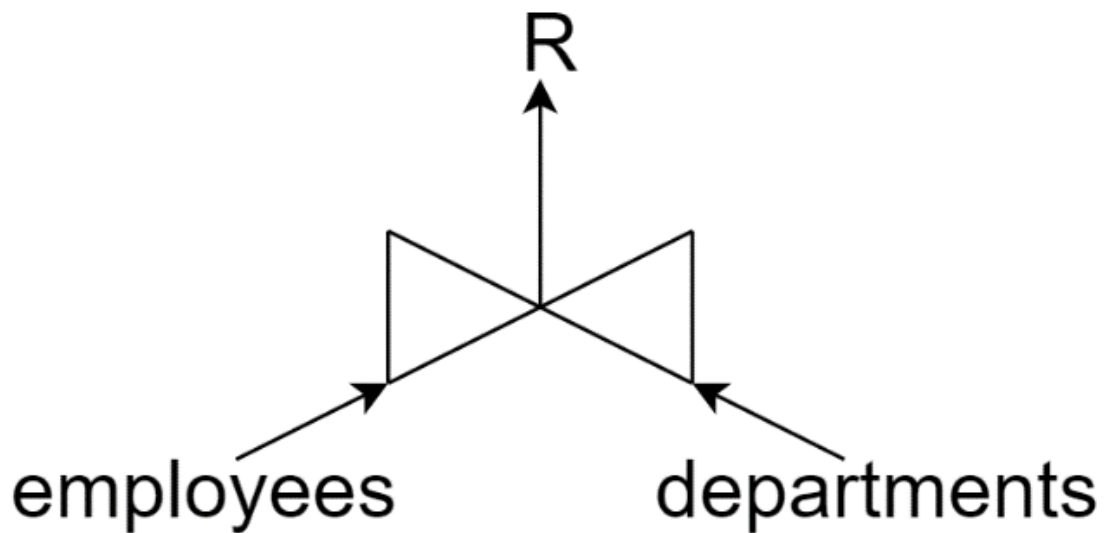
Unde R si S sunt fie tabele, fie relatii rezultate dintr-un alt operator.

Ca exemplu sa afisam datele fiecarui angajat si ale departamentului in care lucreaza.

Expresie algebrica:

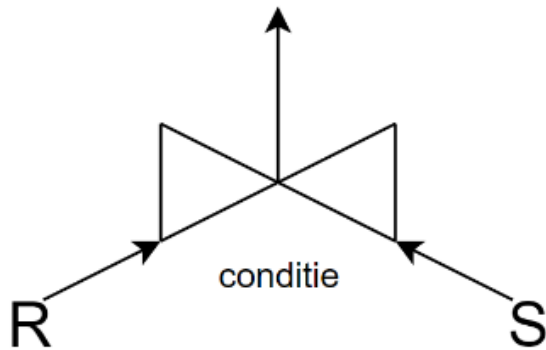
$R = \text{JOIN}(\text{employees}, \text{departments})$

Arbore algebric:



## Operatorul $\theta$ -JOIN

Simbol:



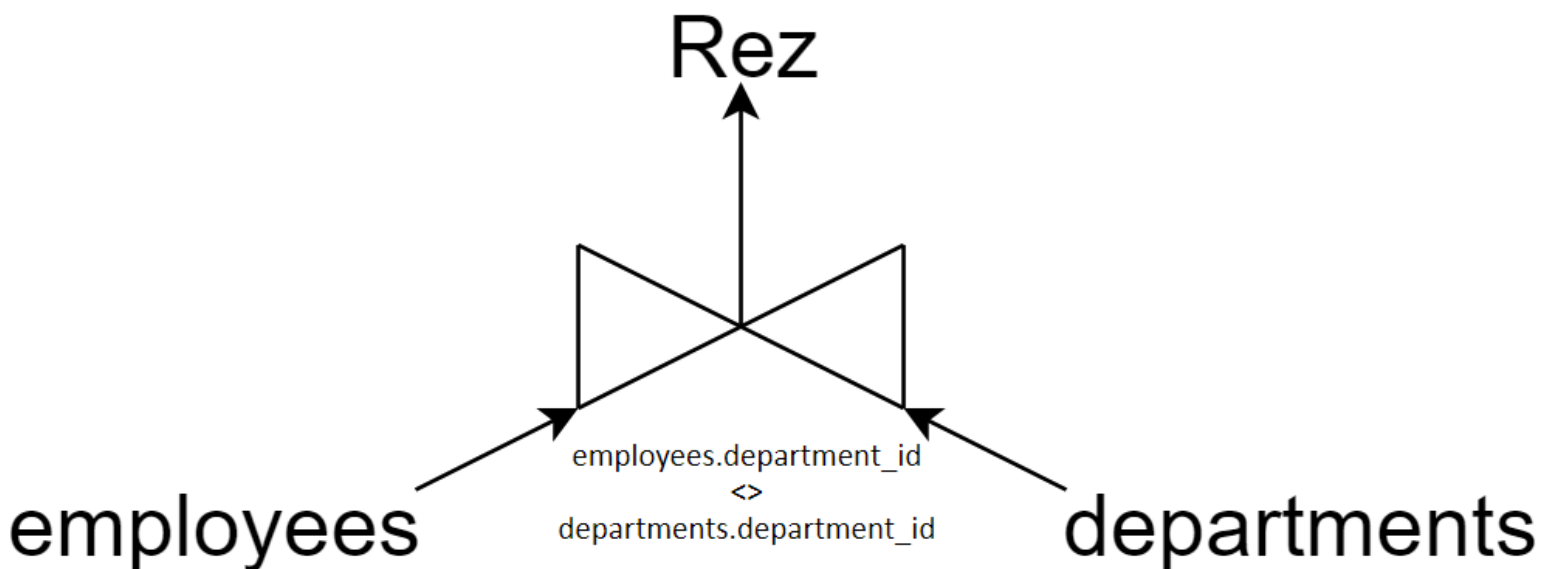
Unde R si S sunt fie tabele, fie relatii rezultate dintr-un alt operator.

Sa exemplificam acest lucru afisand toti angajatii cu toate departamentele mai putin departamentul in care lucreza.

Expresie algebrica:

Rez = **JOIN**(employees, departments, employees.department\_id<>departments.department\_id)

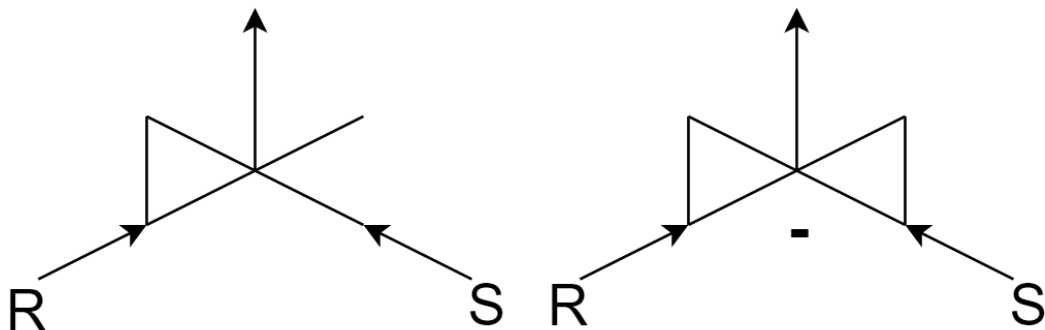
Arbore algebric:





## Operatorul OUTERJOIN

Simbol:



Unde R si S sunt fie tabele, fie relatii rezultate dintr-un alt operator.

Daca operatorul este **LEFT** sau **RIGHT OUTER JOIN** atunci se va folosi simbolul din stanga cu linie lipsa fiind in partea in care se face **OUTER JOIN**. Daca operatorul este **FULL OUTER JOIN** atunci se va folosi simbolul din dreapta.

Ca exemplu sa afisam: toti angajatii si departamentele lor, inclusiv angajatii fara departament; toate departamentele si angajatii ce lucreaza in ele, inclusiv departamentele fara angajati; toti angajatii(inclusiv cei fara departament) cu toate departamentele(inclusiv cele fara angajati).

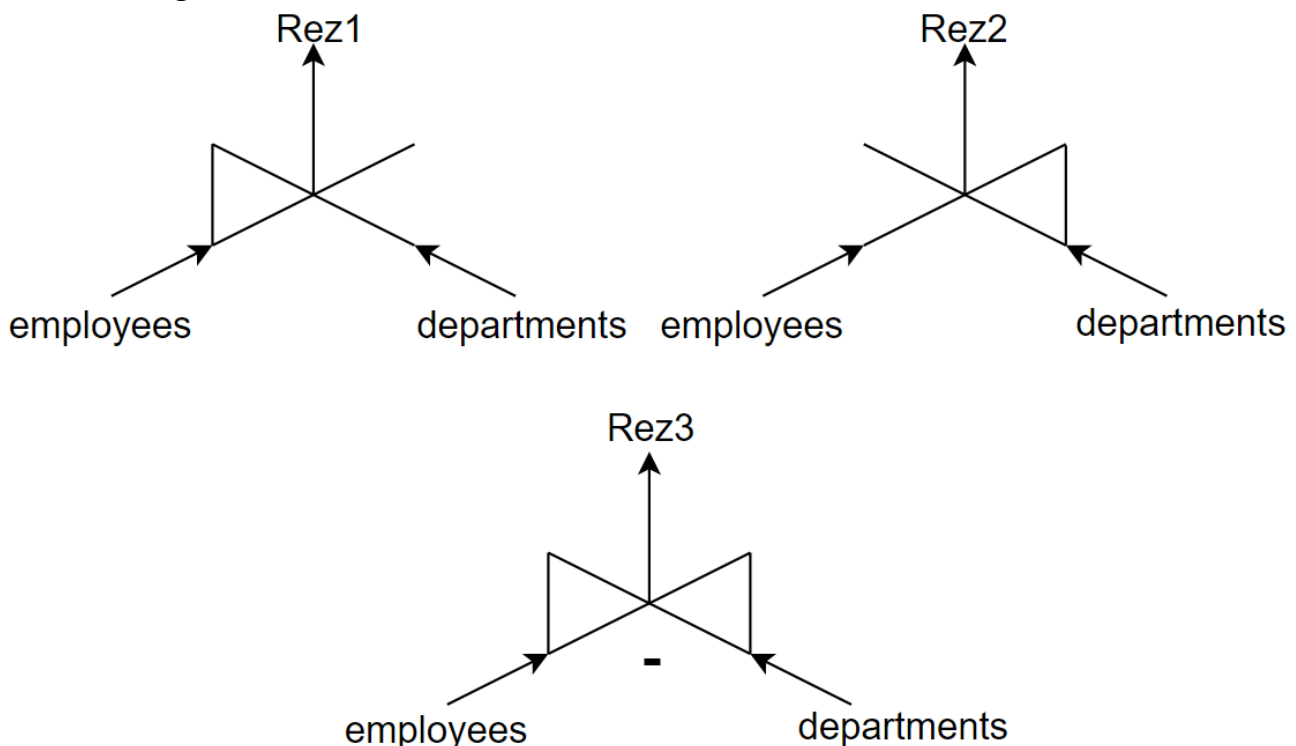
**Expresie algebrica:**

Rez1 = **OUTERJOIN RIGHT**(employees,departments)

Rez2 = **OUTERJOIN LEFT**(employees,departments)

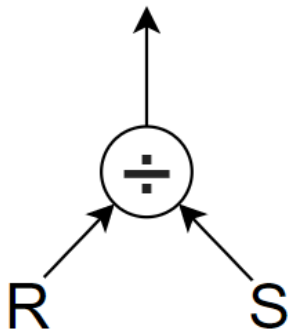
Rez3 = **OUTERJOIN FULL**(employees,departments)

**Arbore algebric:**



## Operatorul DIVISION

Simbol:



Unde R si S sunt fie tabele, fie relatii rezultate dintr-un alt operator.

Sa obtinem codurile salariatilor atasati tuturor proiectelor pentru care s-a alocat un buget de 1000.

Expresie algebrica:

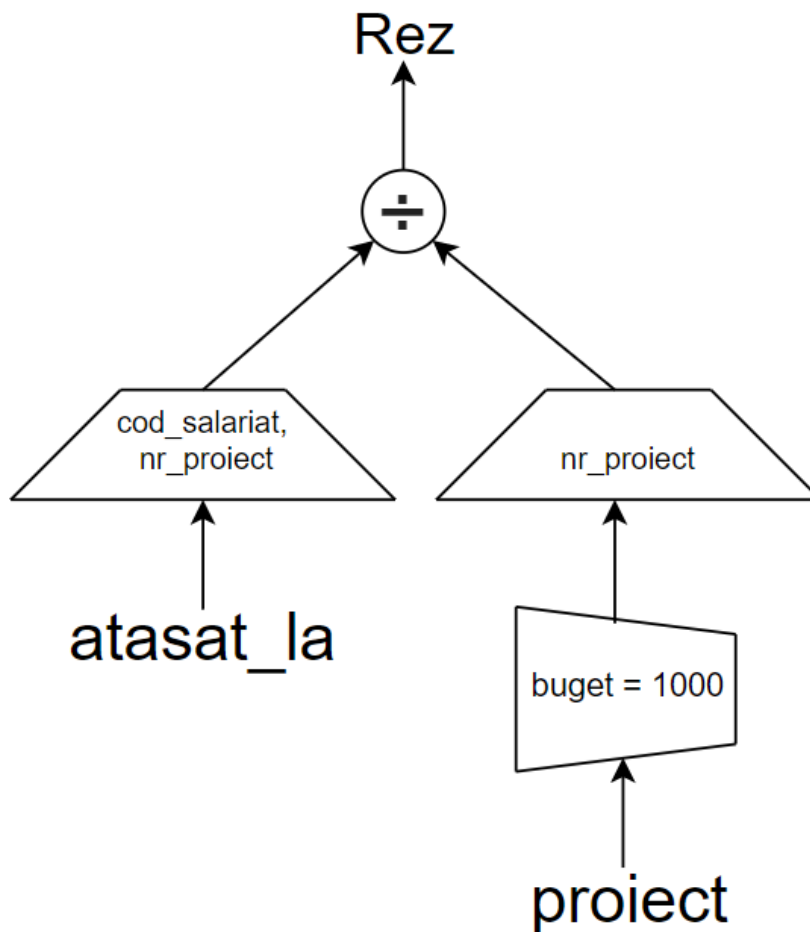
$R = \text{PROJECT}(\text{atasat\_la}, \text{cod\_salariat}, \text{nr\_proiect})$

$S1 = \text{SELECT}(\text{proiect}, \text{buget} = 1000)$

$S2 = \text{PROJECT}(S1, \text{nr\_proiect})$

$\text{Rez} = \text{DIVISION}(R, S2)$

Arbore algebric:



**Mai multe exemple pot fi gasite in drive-ul de anul II, in folder-ul „Exercitii”.**

(<https://drive.google.com/drive/folders/1EdoQG0wErcVYIkCkWFyVeJsmxcQWWsJZ>)

In final sa scriem expresia algebrica si arborele algebric pentru urmatoarea cerere: sa se afiseze numele si prenumele angajatilor ce il au ca manager pe „Steven King” si au fost angajati in anul 2000.

**Expresia algebrica:**

R1 = **SELECT**(employees, first\_name = 'Steven')

R2 = **SELECT**(employees, last\_name = 'King')

R3 = **INTERSECT**(R1,R2)

R4 = **PROJECT**(R3, employee\_id)

S1 = **JOIN**(employees, R4, employees.manager\_id = R4.employee\_id)

S2 = **SELECT**(S1, TO\_CHAR(hire\_date,'YYYY') = '2000')

Rez = **PROJECT**(S2, first\_name, last\_name)

Arborele algebric:

