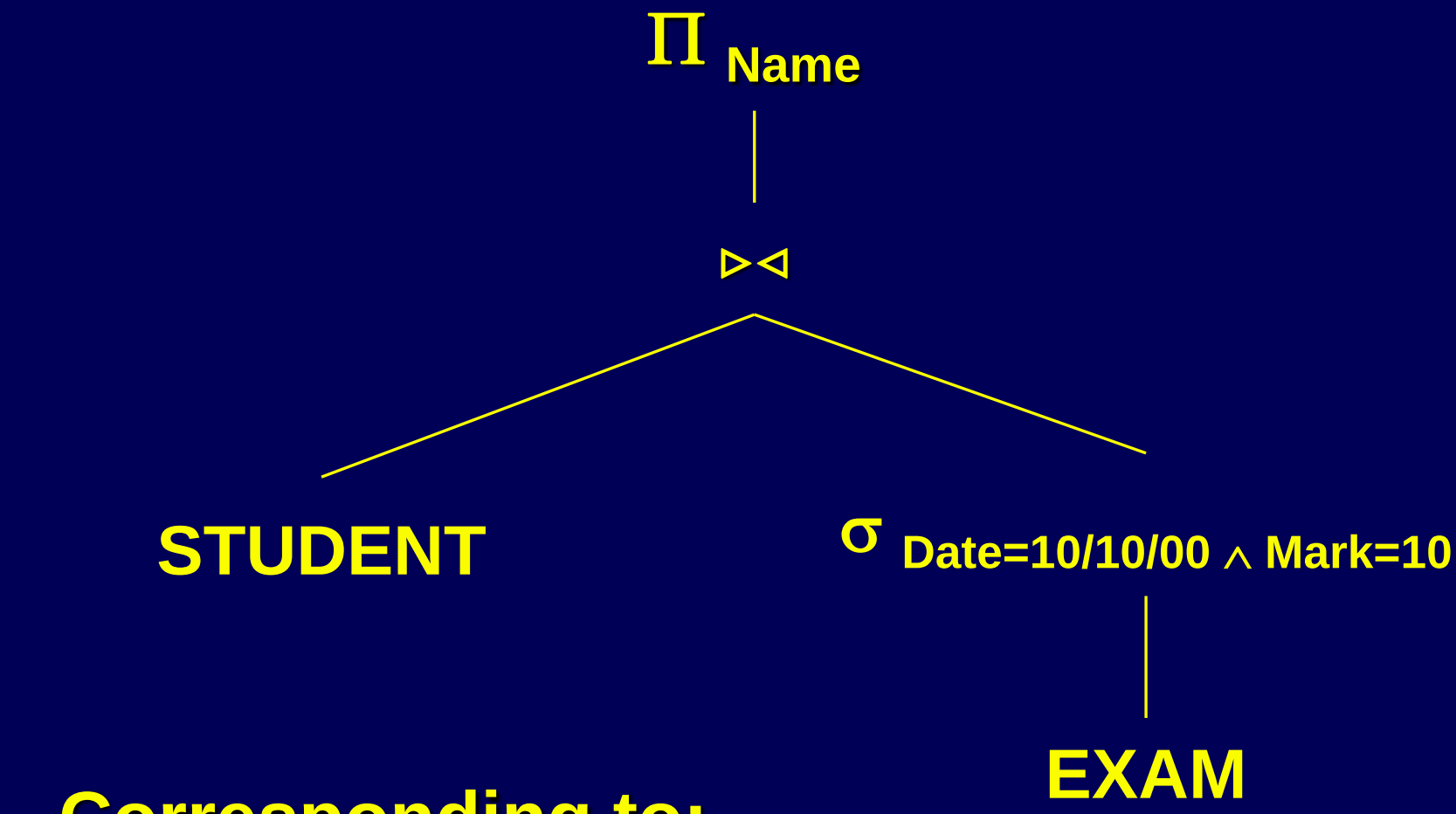
Query Optimization

Queries and Trees

- Every query of the RA can be graphically represented by a tree, which states the sequence of evaluation of operators.
- Every operator is mapped as a node:
  - unary operators have one incoming branch and one outgoing branch;
  - binary operators have two incoming branches and one outgoing branch.

Tree

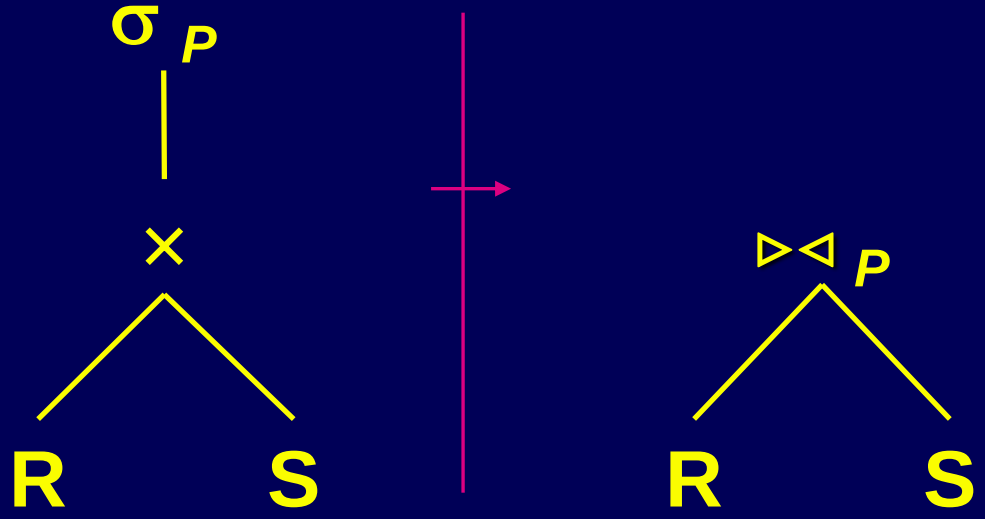


Corresponding to:

$$\Pi_{Name}\ STUDENT \bowtie \sigma_{Date=10/10/00 \wedge Mark=10}\ EXAM$$

Equivalence Transformations for RA

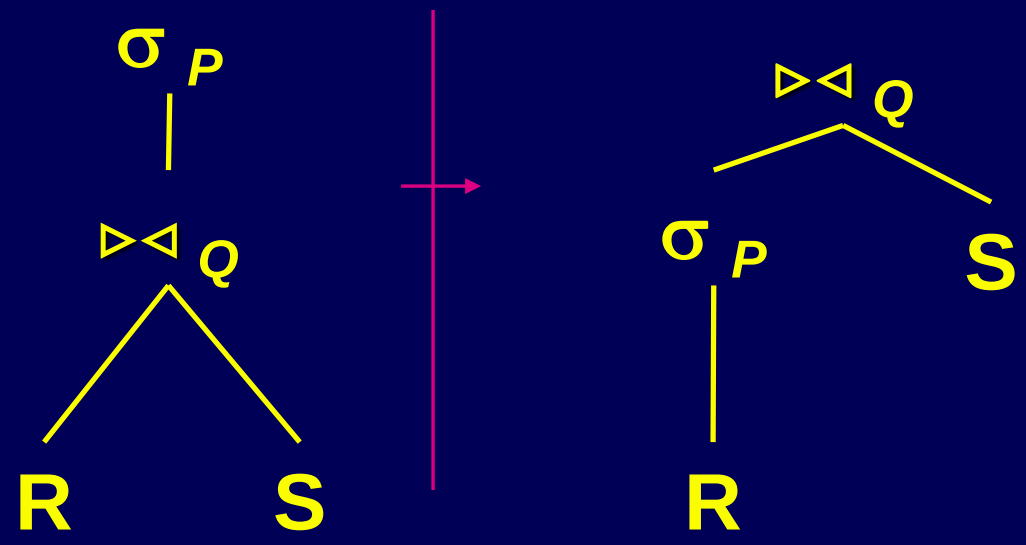
1) Removal of Cartesian products



If *p* is a conjunction of predicates such as ATTR *comp* ATTR

Equivalence Transformations for RA

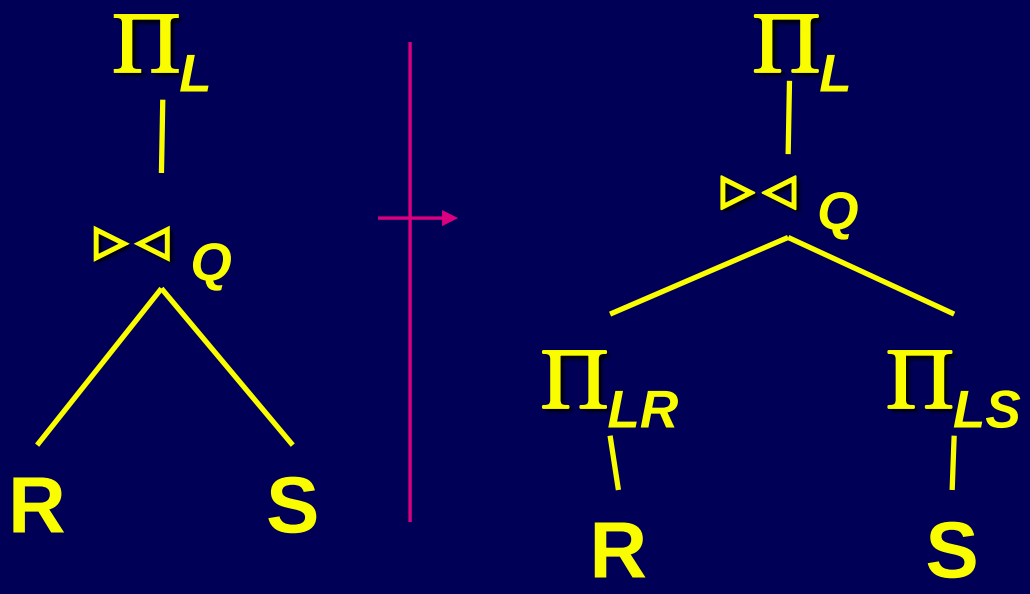
2) Push of the selection wrt join



If *p* is a predicate applicable for the attributes of R, ONLY.

Equivalence Transformations for RA

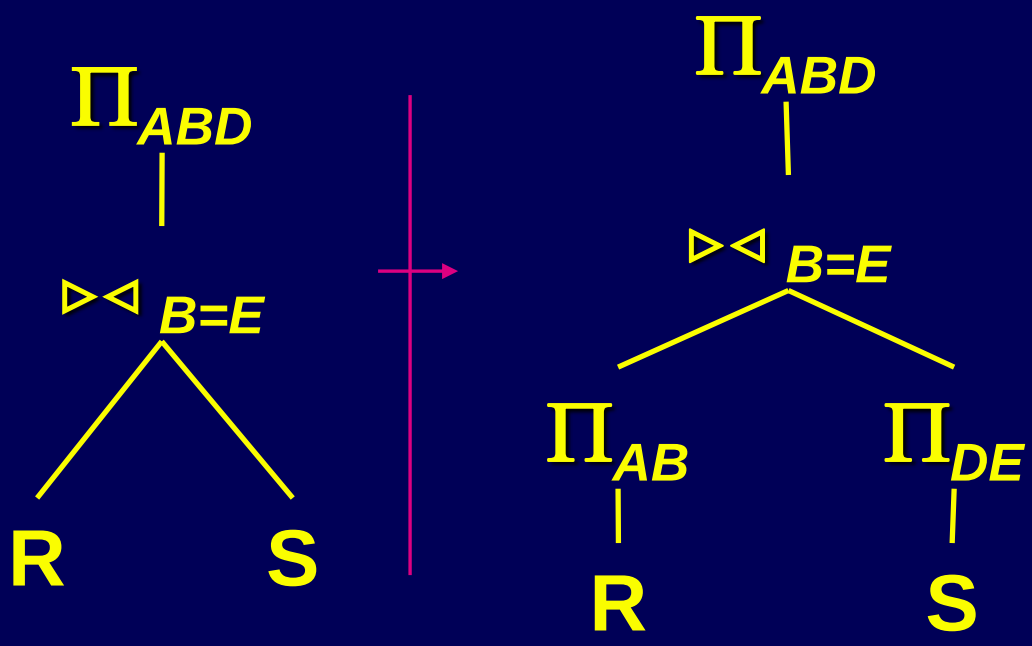
3) Push of the project operator wrt join



JR and JS are the attributes of R and S to evaluate *Q*
LR = *L* - schema(S) + *JR*, *LS* = *L* - schema(R) + *JS*

Example

3) Push of the project operator wrt join

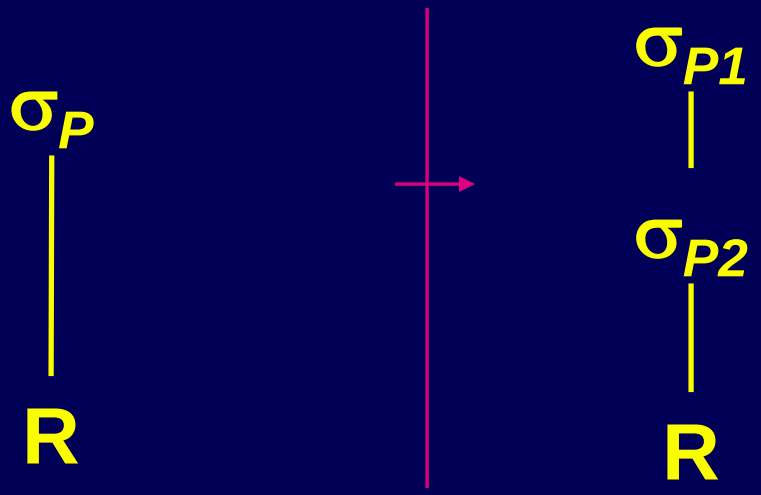


R(A,B,C)
S(D,E,F,G)

L=ABD
JR=B
JS=E
LR=AB
LS=DE

Equivalence Transformations for RA

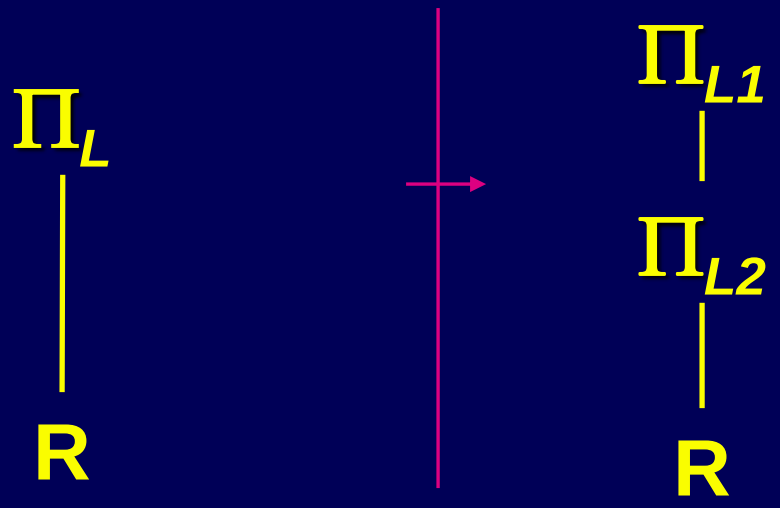
4) Idempower of the select operator



$P = P1 \wedge P2$

Equivalence Transformations for RA

5) Idempower of projection

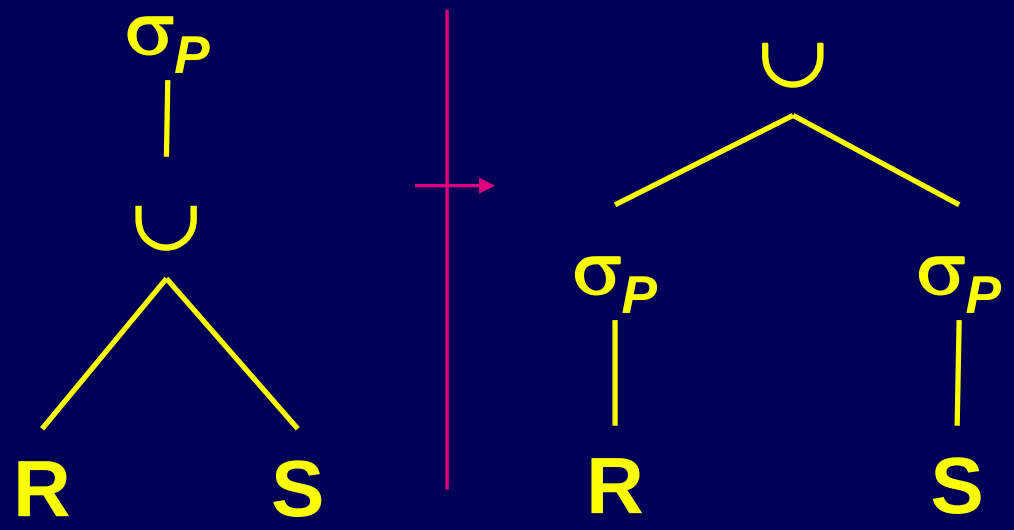


$L1 = L$

$L2 \supseteq L$

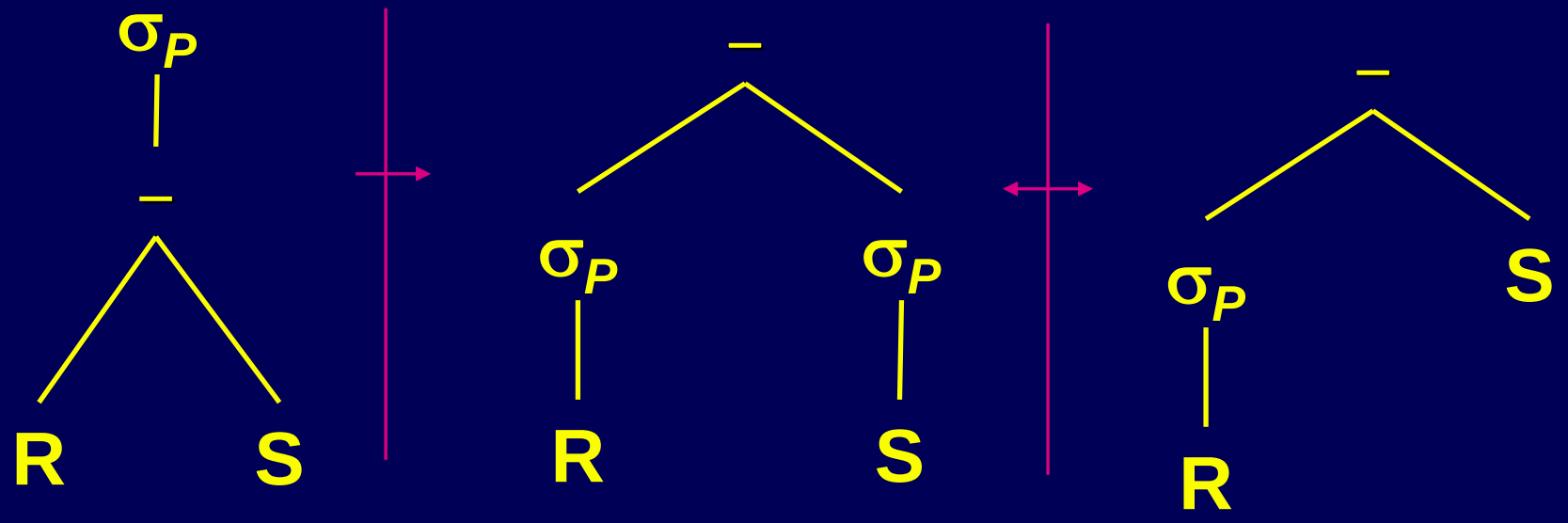
Equivalence Transformations for RA

6) Push of the select operator wrt Union operator



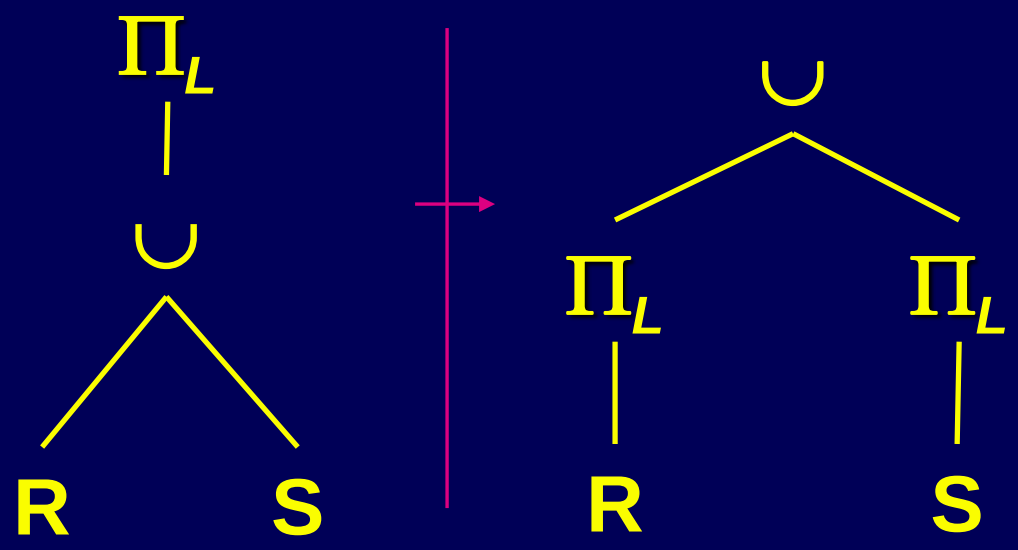
Equivalence Transformations for RA

7) Push of the select operator wrt Difference operator



Equivalence Transformations for RA

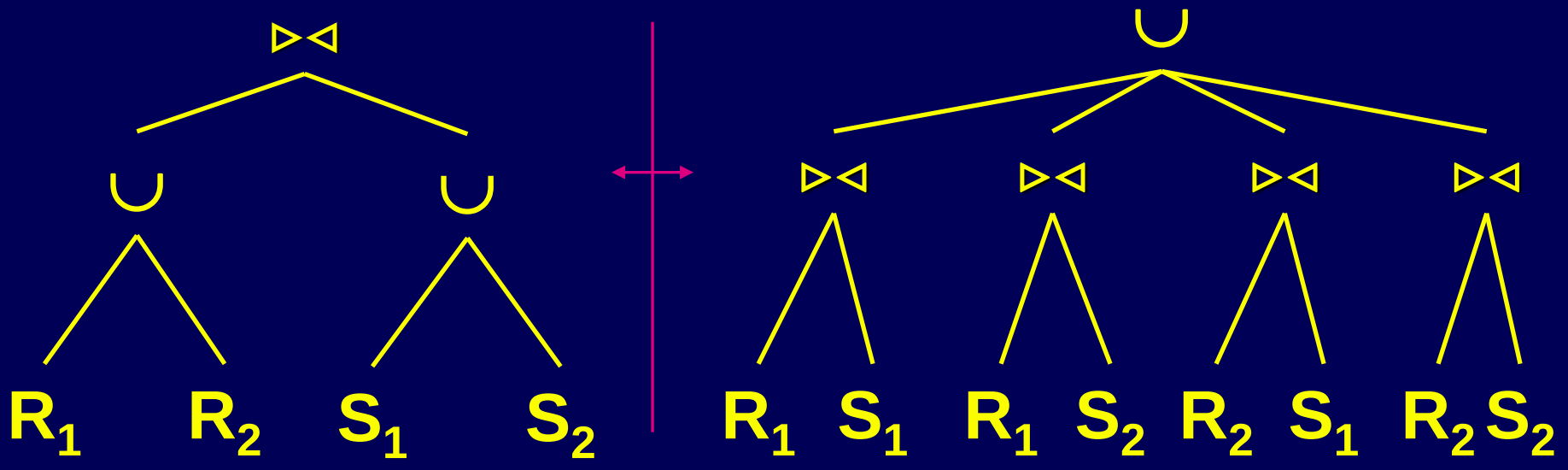
8) Push of the project operator wrt Union



Warning: push of the project operator wrt difference or intersection does NOT apply.

Equivalence Transformations for RA

9) Commutative properties of join and Union



Useful Formulae

$R \triangleright\!\triangleleft R = R$

$R \cup R = R$

$R - R = \emptyset$

$R \triangleright\!\triangleleft \sigma_P R = \sigma_P R$

$R \cup \sigma_P R = R$

$R - \sigma_P R = \sigma_{\neg P} R$

$\sigma_{P1} R \triangleright\!\triangleleft \sigma_{P2} R = \sigma_{P1 \wedge P2} R$

$\sigma_{P1} R \cup \sigma_{P2} R = \sigma_{P1 \vee P2} R$

$\sigma_{P1} R - \sigma_{P2} R = \sigma_{P1 \wedge \neg P2} R$

$\sigma_P \emptyset = \emptyset$

$\Pi_L \emptyset = \emptyset$

$R \cup \emptyset = R$

$R - \emptyset = R$

$\emptyset - R = \emptyset$

$R \cap \emptyset = \emptyset$

$R \times \emptyset = \emptyset$

$R \triangleright\!\triangleleft \emptyset = \emptyset$

Algebraic Optimization

- Among all the equivalent expressions, choose the most efficient ones.
- Informal criterion: minimize size of intermediate results.
- Approach:
  - use push when possible (2, 3, 6, 7, 8);
  - use idempower (4, 5) to separate individual selections/projections.

Example

Given the following schema:

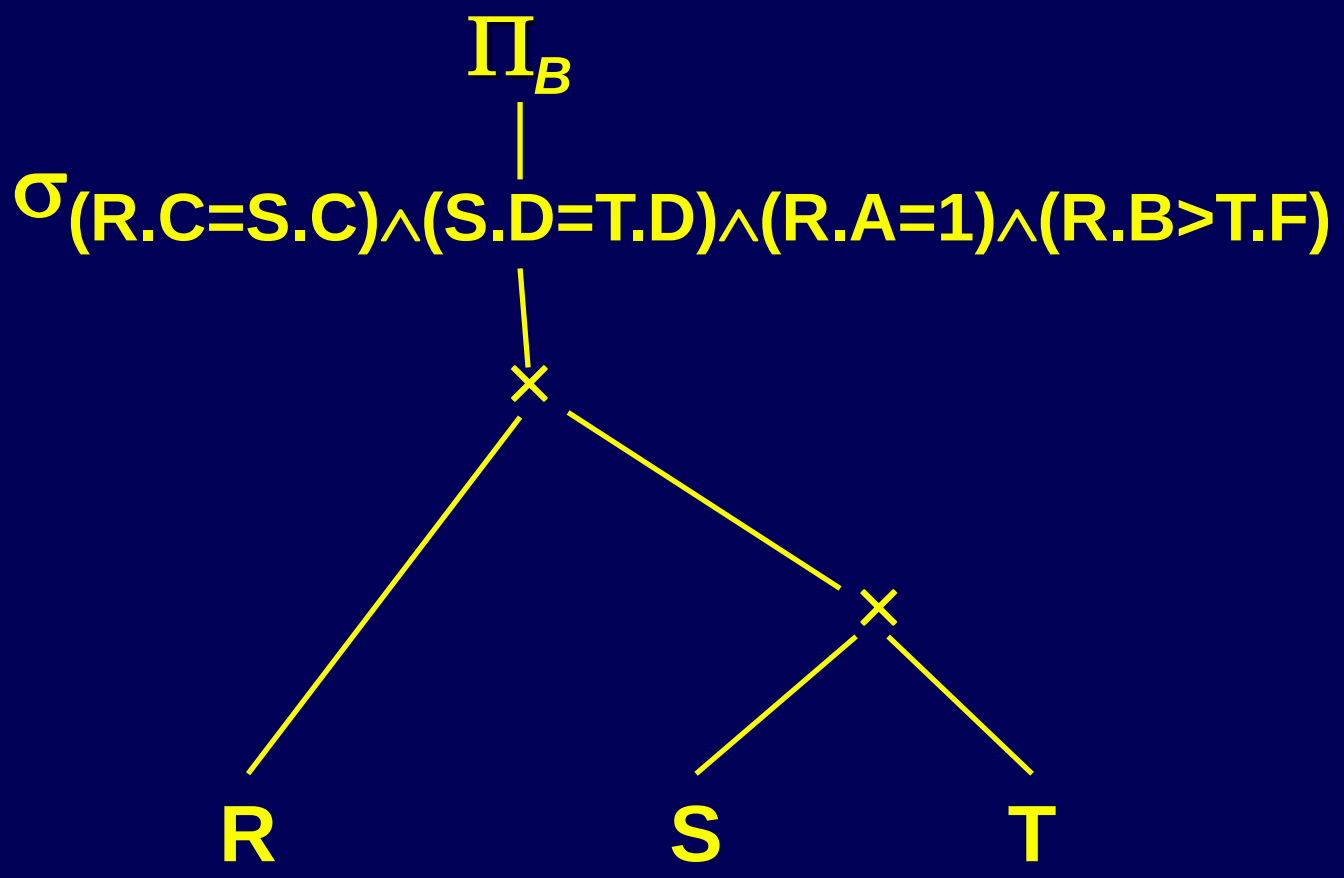
R(A,B,C)
S(C,D,E)
T(D,E,F,G)

Optimize the following algebraic expression:

$\Pi_B \sigma_{(R.C=S.C)\wedge(S.D=T.D)\wedge(R.A=1)\wedge(R.B>T.F)} R \times S \times T$

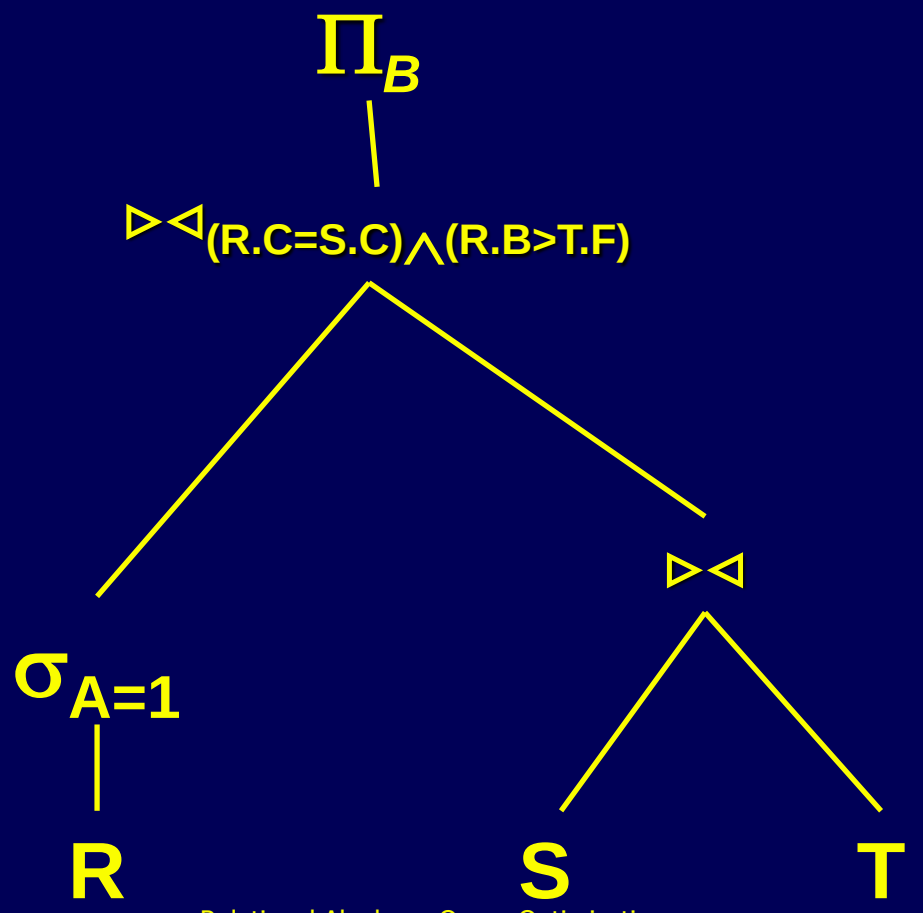
Example

By a tree graphical representation, we obtain:



Example

By the transformations, we obtain:

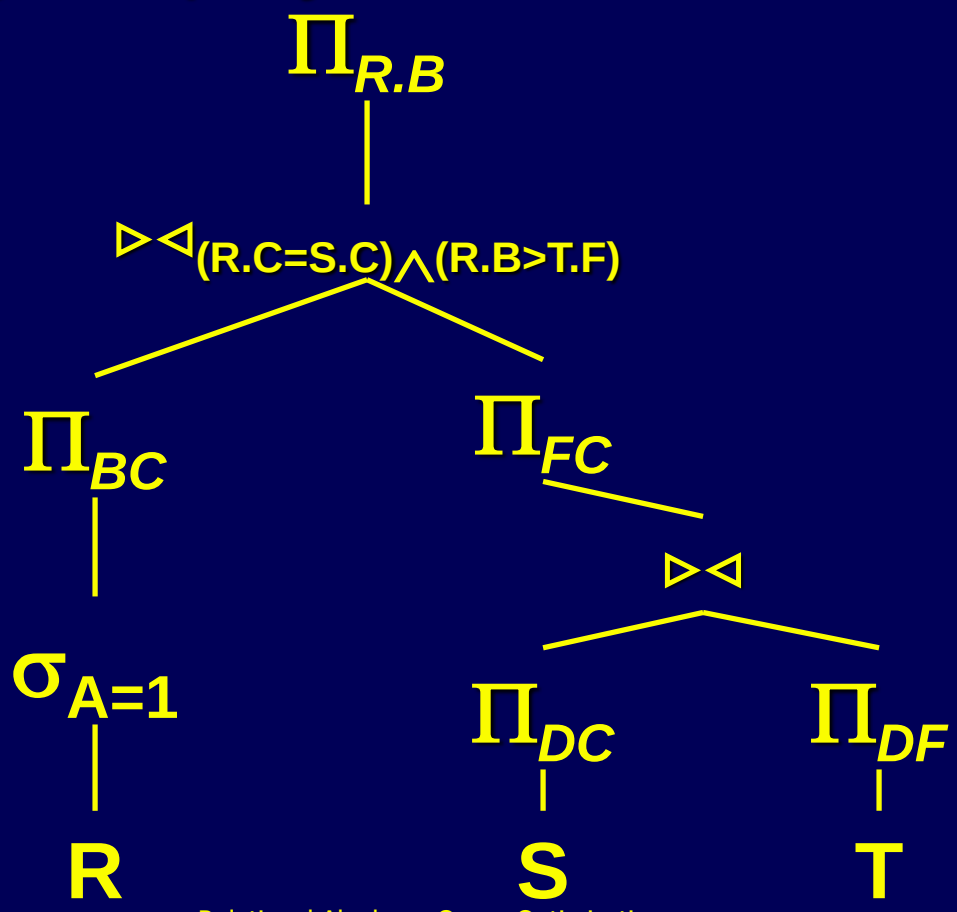


we omit (S.D=T.D)

push of the selection wrt join

Example

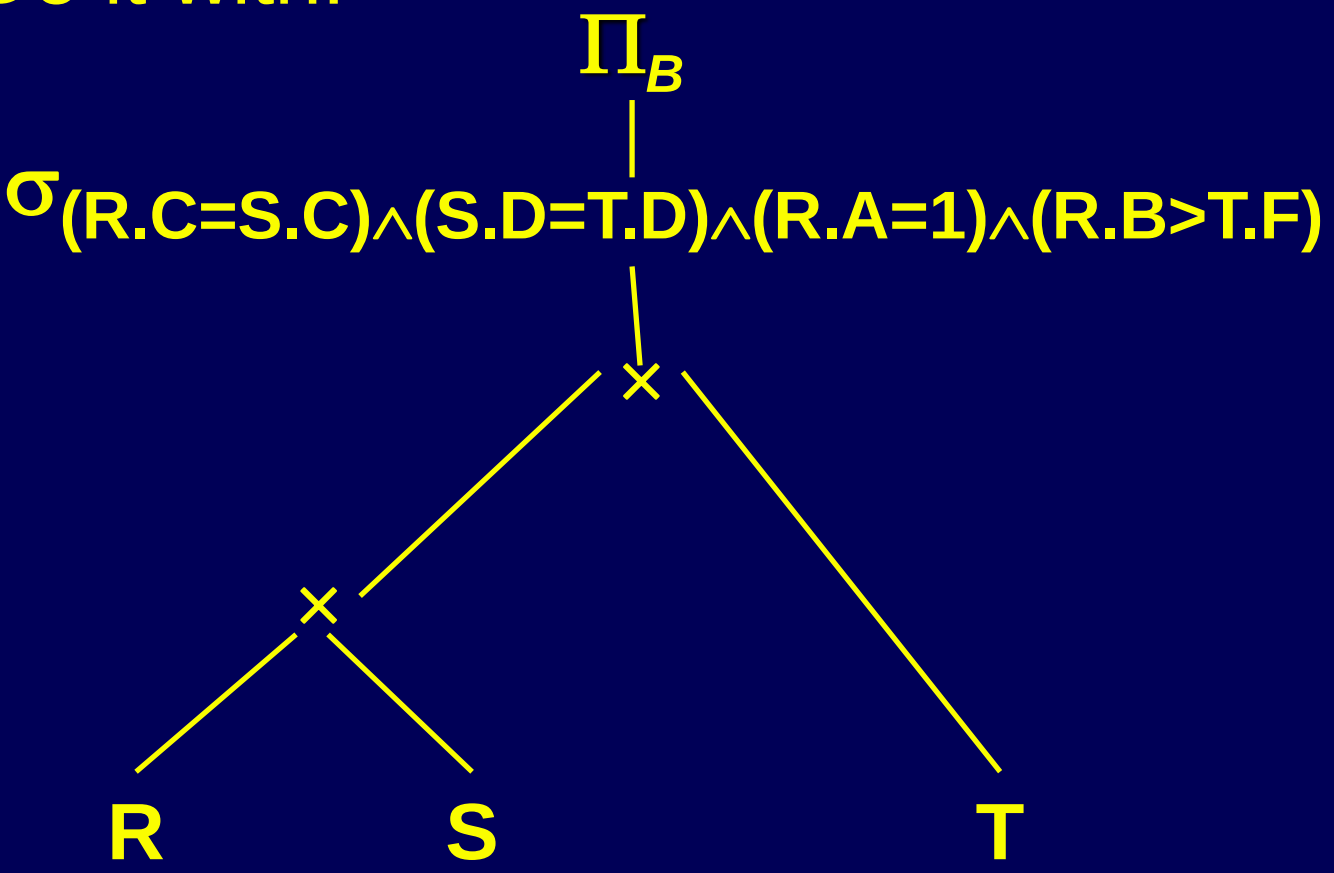
By adding the projections, we obtain:



push of the project wrt join

Exercise

- Do it with:



Exercise

1) Given the schema: R(A,B)
S(A,B)

Optimize the following algebraic expression:

$\sigma_{(S.A=R.A)\wedge(R.A>2)\wedge(S.A=1)}\ R \times S$

The result is an empty relation, because the predicate is a contradiction!!!!

Exercise

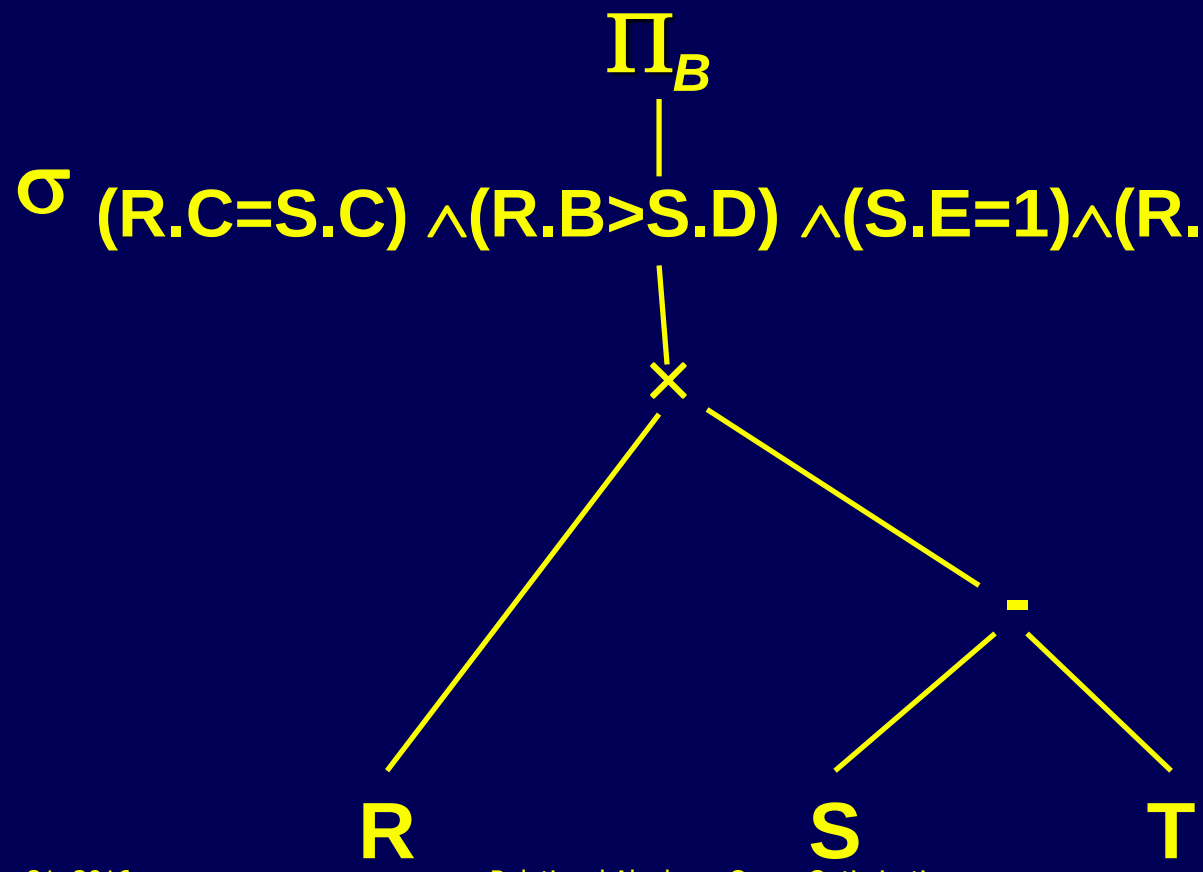
2) Given the schema: R(A,B,C)
S(C,D,E)
T(C,D,E)

Optimize the following algebraic expression:

$\Pi_B \sigma_{(R.C=S.C) \wedge (S.E=1) \wedge (R.B=2) \wedge (R.B>S.D)} (R \times (S - T))$

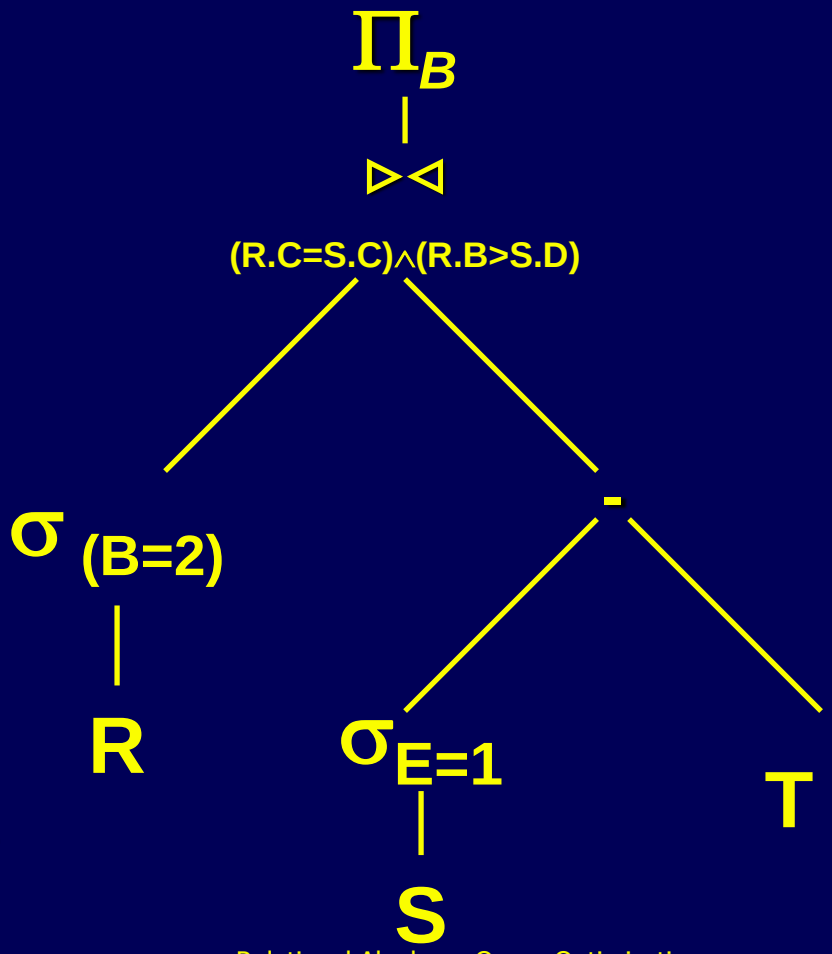
Example

- By a tree graphical representation, we obtain:



Example

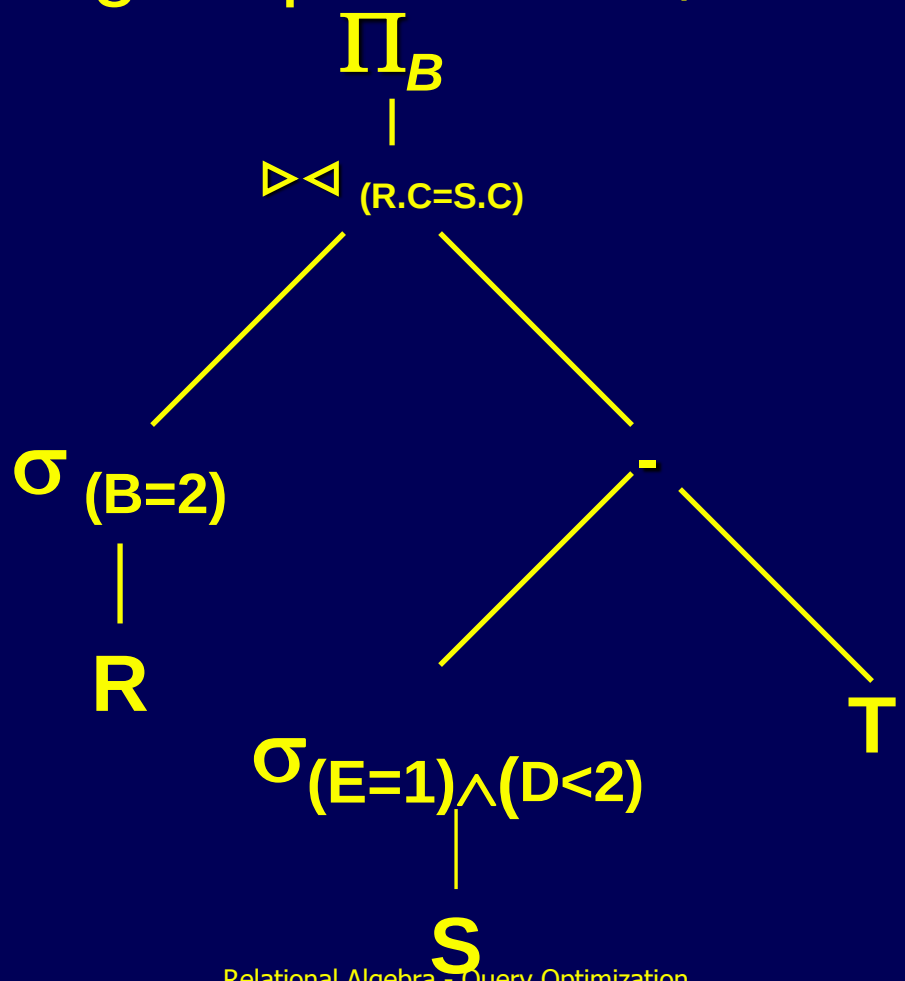
- By the transformation, we obtain



push of the select wrt join

Example

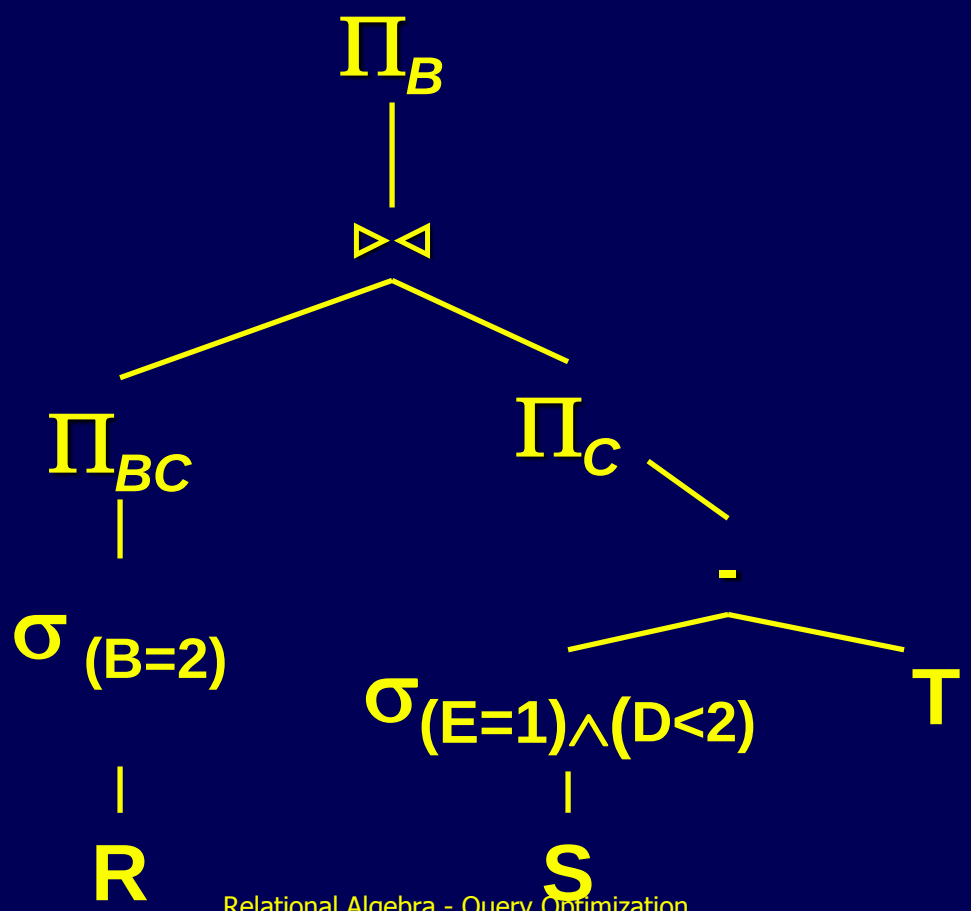
- Reasoning on predicates, we obtain:



if B=2, then
D < 2

Example

- Adding the projects, we obtain:



push of the project wrt join

Comments

- About the last query:
  - a further push of Π_c over the difference S – T is possible;
  - it can be easily observed that the final result is either the empty set or a set made of one tuple with the value 2.
    - This latter if R has at least one tuple where B=2 and the join between R and (S – T) returns at least one tuple.