Reznichenko Valery
Organization of data and knowledge bases

Lecture 6. Relational algebra

National Aviation University
Computer Science Faculty
Department of Software Engineering

CONTENTS

Query languages

Query language is a language that allows to extract data from database.

Language categories:

- procedural (HOW to receive)
- nonprocedural (WHAT to receive)

Formal languages:

- relational algebra
- relational calculus (tuple-oriented and domain-oriented)

Formal languages are basis for creation of DB query languages (Alpha, QUEL, QBE, SQL)

Relational algebra closure and properties of binary operations

Algebra = data (of the defined type) + operations.
Algebra is closured if result of any operation are the same type as a data in argument. Closure property allows to embed operations in each other.
Relational algebra = relations + operations.
Relational algebra closured.

Property of binary relations:

- Operation ϕ is ***commutative*** if $A \phi B = B \phi A$
- Operation ϕ is ***associative*** if $(A \phi B) \phi C = A \phi (B \phi C)$
- Operation ϕ is ***distributive*** with respect with other operation θ, if $A \phi (B \theta C) = (A \phi B) \theta (A \phi C)$

Relational algebra operations

Basic operations:

- set-theoretic (union, intersection, difference)
- projection
- selection
- cartesian product,
- join
- division

Additional operations

- assignment
- renaming
- generalized projection
- outer join
- …

Set-theoretic operations

Two relations R and S are (***union***) ***compatible*** if:

- R and S have the same arity/ that is they have the same number of attributes.
- Domains of corresponding attributes are compatible (the 1-st attribute of R is defined on the same domain as the 1-st attribute of S and so on) .

Set-theoretic operations require compatibility of their operands

Union operation

Union of two compatible relations R and S with schemas R(A) and S(A), where A is a set of attributes, is the relation T with schema T(A) that contains tuples of both relations, but without duplicates.

$$T(A) = R(A) \cup S(A) = \{t \mid t \in R \vee t \in S\}$$

The operation commutative, associative and distributive with respect to the intersection .

Example:

R

A	B
a_1	b_1
a_1	b_2
a_2	b_3

S

A	B
a_1	b_1
a_2	b_1

T = R ∪ S

A	B
a_1	b_1
a_1	b_2
a_2	b_1
a_2	b_3

Difference operation

Difference of two compatible relations R and S with schemas R(A) and S(A), where A is a set of attributes, is the relation T with schema (A) that contains only those tuples of the relation R, which are not present in the relation S.

$$T(A) = R(A) - S(A) = \{t \mid t \in R \ \& \ t \notin S\}$$

The operation not commutative and associative and distributive.

Example:

R

A	B
a_1	b_1
a_1	b_2
a_2	b_3

S

A	B
a_1	b_1
a_2	b_1

T = R — S

A	B
a_1	b_2
a_2	b_3

Note: Relational algebra do not use set-theoretic complement operation!!!

Intersection operation

Intersection of two compatible relations R and S with schemas R(A) and S(A), where A is a set of attributes, is the relation T with schema T(A) that contains tuples which simultaneously present in both relations.

$$T(A) = R(A) \cap S(A) = \{t \mid t \in R \ \& \ t \in S\}$$

The operation commutative, associative and distributive with respect to union.

Example:

R

A	B
a_1	b_1
a_1	b_2
a_2	b_3

S

A	B
a_1	b_1
a_2	b_1

$T = R \cap S$

A	B
a_1	b_1

Intersection is expressed via difference: $R \cap S = R - (R - S)$

Projection operation

Projection of the relation R with schema R(A), where A is a set of attributes, with respect to the set of attributes B, where $B \subset A$, is a such relation S with the schema S(B) that contains tuples of the relation R by deleting from them values that belong attributes A - B.

Projection is denoted as: R[B] или $\pi_{\mathbf{B}}(R)$

$S(B) = R[B] = \{t[B] \mid t \in R\}$

Example:

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_1
a_2	b_3	c_1
a_2	b_4	c_2

R[A, C]

A	C
a_1	c_1
a_2	c_1
a_2	c_2

Note: Duplicate tuples are deleted

θ-comparability of attributes and tuples

Let's θ is any of the following comparison operators : =, ≠, < ≤, >, ≥. The attributes A and B the same or different relations are ***θ-comparable***, if for any values $a \in A$ and $b \in B$ expression $a\ \theta\ b$ is defined.

Set of attributes M = (A1,…, Ak) and N = (B1,…,Bn) are ***θ-comparable***, if k = n and (Ai, Bi) are θ-comparable. In this case expression M θ N means the following:

M θ N = (A1 θ B1) & … & (Ak θ Bk)

If t is a relation tuple, that contains sets of θ-comparable attributes M and N, then the notation t[M] θ t[N] means the following:
(t[A1] θ t[B1]) & … & (t[Ak] θ t[Bk]).

Selection (restriction) operation

Let's M and N are sets of θ-comparable attributes of the relation R with schema R(A). Then ***selection*** (***restriction***) of the relation R on condition M θ N, denoted as R [M θ N], is such relation T with schema T(A), which tuples satisfy the condition t[M] θ t[N].

$$T(A) = R[M\ \theta\ N] = \{t \mid t \in R\ \&\ t[M]\ \theta\ t[N]\}$$

One of the attributes set M or N may be constant.

Example:

R

A	B
a_1	b_1
a_1	b_2
a_2	b_3

R[A=a1]

A	B
a_1	b_1
a_1	b_2

The operation also has the following notation: $\sigma_{M\ \theta\ N}(R)$

Cartesian product

Cartesian product of two rations R and S with schemas R(A) and S(B) (A and B are sets of attributes), denoted as R(A)×S(B), is a relation T of degree n+m with schema T(A, B) that contains all possible concatenations of tuples of relations R and S:

$$T(A,B) = R(A) \times S(B) = \{(t1,t2) \mid t1 \in R \;\&\; t2 \in S\}$$

The operation is commutative and associative and distributive with respect of union and intersection.

Example:

R

A	B
a_1	b_1
a_1	b_2
a_2	b_3

S

C	D
c_5	d_3
c_4	d_7

T = R x S

A	B	C	D
a_1	b_1	c_5	d_3
a_1	b_1	c_4	d_7
a_1	b_2	c_5	d_3
a_1	b_2	c_4	d_7
a_2	b_3	c_5	d_3
a_2	b_3	c_4	d_7

Join operation

Let us M and N are sets of θ-comparable attributes. ***Join*** of the relations R and S with schemas R(A,M) and S(N,B) on a condition M θ N, denoted as R[M θ N]S, is a relation T with a schema T(A,M,N,B) which tuples are concatenation only those tuples of R and S, for which sets of attributes N and M satisfy condition M θ N.

$$T = R[M\ \theta\ N]S = \{(t1, t2) \mid t1 \in R \wedge t2 \in S \wedge t1[M]\ \theta\ t2[N]\}$$

The operation commutative and associative

Example:

R

A	M
a_1	n_1
a_2	n_1
a_2	n_3

S

N	B
n_1	b_2
n_3	b_4
n_3	b_5

R [M=N]S

A	M	N	B
a_1	n_1	n_1	b_2
a_2	n_1	n_1	b_2
a_2	n_3	n_3	b_4
a_2	n_3	n_3	b_5

The operation is also denoted as: $R \bowtie_{M\ \theta\ N} S$

Join is expressed via the product and the selection:

$$R \bowtie_{M\ \theta\ N} S = \sigma_{M\ \theta\ N}(R \times S)$$

Join and natural join

Join on a condition of equality (=) is called as ***equi-join*** .

Natural join is a join on equality condition by attributes with the same names with deleting in the result relation one of the compared set of attributes.

Natural join is denoted by the symbol * (for example R*S).

Example:

R

A	B	C
a_1	b_1	c_2
a_2	b_1	c_3
a_2	b_3	c_5

S

B	C	D
b_1	c_2	d_7
b_3	c_5	d_4
b_3	c_5	d_2

R[B,C=B,C]S

A	B	C	B	C	D
a_1	b_1	c_2	b_1	c_2	d_7
a_2	b_3	c_5	b_3	c_5	d_4
a_2	b_3	c_5	b_3	c_5	d_2

R*S

A	B	C	D
a_1	b_1	c_2	d_7
a_2	b_3	c_5	d_4
a_2	b_3	c_5	d_2

Semijoin

Semijoin is join of two relations with deleting from the resulting relation attributes of one of the joined relations.

Semijoin is denoted as: R[M θ N)S

$$R[M\ \theta\ N)S = \{(t1) \mid t1 \in R \wedge t2 \in S \wedge t1[M]\ \theta\ t2[N]\}$$

Semijoin is expressed via the join and projection in such a way:

R[M **θ** N)S = (R[M **θ** N]S)[A] – where A is a set of attributes of the relation R

Example:

R

A	B	C
a_1	b_1	c_2
a_2	b_1	c_3
a_2	b_3	c_5

S

B	C	D
b_1	c_2	d_7
b_3	c_5	d_4
b_3	c_5	d_2

R[B,C=B,C)S

A	B	C
a_1	b_1	c_2
a_2	b_3	c_5

Image of the tuple

Image of the relation R(M,N) with respect the tuple t1 ∈ R[M], that is denoted as $I_{t1}(R)$, is the set of such tuples t2 ∈ R[N], that concatenation of tuples (t1,t2) belongs to the relation R.

$$I_{t1 \in R[M]}(R) = \{(t2) \mid t2 \in R[N] \wedge (t1,t2) \in R\}$$

Examples:

R

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_3	c_2
a_2	b_1	c_4

$I_{a1 \in R[A]}(R)$

B	C
b_1	c_1
b_1	c_2
b_3	c_2

$I_{a2 \in R[A]}(R)$

B	C
b_1	c_4

$I_{(a1,b1) \in R[A,B]}(R)$

C
c_1
c_2

$I_{c2 \in R[C]}(R)$

A	B
a_1	b_1
a_1	b_3

Division operation (1)

Division of two relations R(M,N) and S(K,L) with respect of set of compatible attributes N and M, denoted as R [N ÷ K] S, is a relation T(M) with such tuples t ∈ R[M] whose images $I_t(R)$ contain all tuples of the projection S[K].

$$T(M) = R[N \div K]S = \{t \mid t \in R[M] \ \& \ I_t(R) \supseteq S[K]\}$$

The operation allows to formulate queries like this:

- Output teachers that teach lectures of ALL types.

Teacher	Set A All types of lectures, that has a teacher	⊇	Set B All possible types of lectures

Division operation(2)

Example:

R

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_3	c_2
a_2	b_1	c_4

S

C	B
c_1	d_1
c_1	d_2
c_2	d_1
c_2	d_2

S[C]

C
c_1
c_2

R[C÷C]S

A	B
a_1	b_1

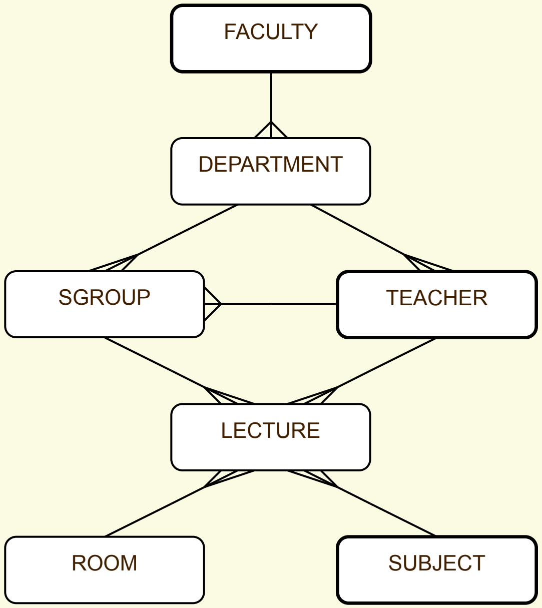
Division operation is expressed by other operations of RA:

R[N÷K]S = R[M] – ((R(M) x S[K]) – R)[M]

Division operation is not commutative and associative

Example of DB for RA queries

```
FAC (FNo, Name, Dean, Bld, Fund)
DEP (DNo, FNo, Name, Head, Bld, Fund)
TCH (TNo, DNo, Name, Post, Tel, Salary, Comm)
GRP (GNo, DNo, Course, Num, Quantity, CurNo)
SBJ (SNo, Name)
ROM (RNo, Num, Building, Seats)
LEC (TNo, GNo, SNo, RNo, Type, Day, Week)
```



Examples of queries in RA (1)

Projection: Output list jf teacher names and posts:

`TCH[Name, Post]` $\pi_{Name,Post}(TCH)$

Selection: Output information about CSF faculty:

`FAC[Name = 'CSF']` $\sigma_{Name='CSF'}(FAC)$

Join: Output information about faculries and their departments:

`FAC[FNo = FNo]DEP` $FAC \bowtie_{FNo=FNo} DEP$

Examples of queries in RA (2)

Composition of join, selection and projection

1) Outputs names of faculties and their departments

`(FAC[FNo = FNo]DEP) [FAC.Name, DEP.Name]`

$$\pi_{FAC.Name,\ DEP.Name}(FAC \bowtie_{FNo=FNo} DEP)$$

2) Outputs names of faculties with fund > 1000 and their departments :

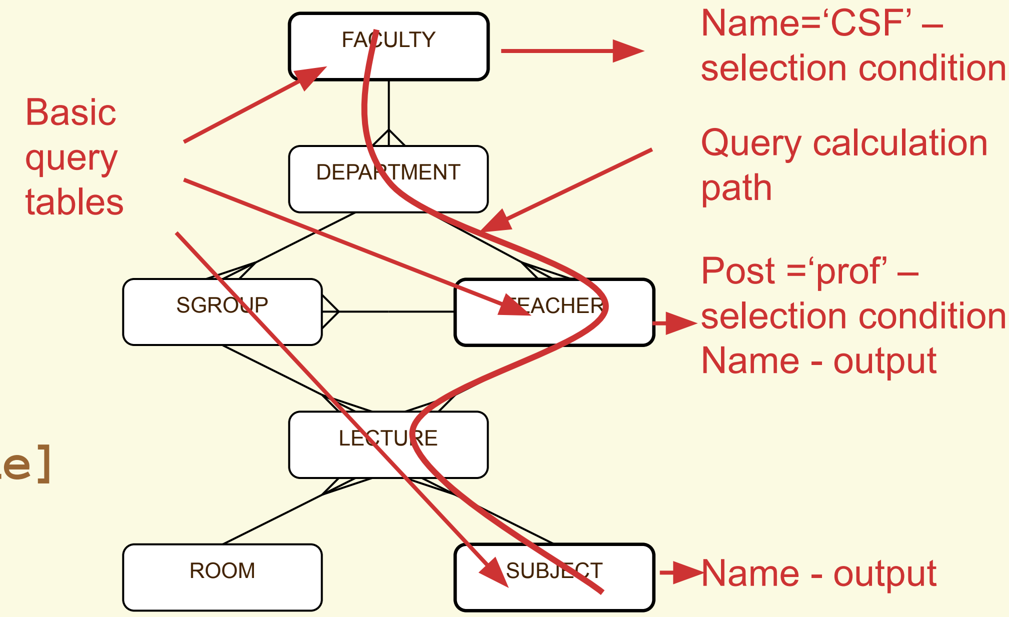
`((FAC[Fund > 1000])[FNo = FNo]DEP) [FAC.Name, DEP.Name]`

$$\pi_{FAC.Name,\ DEP.Name}((\sigma_{Fund\ >1000}(FAC)) \bowtie_{FNo=FNo} DEP)$$

Examples of queries in RA (3)

Outputs names of professors from CSF faculty and subjects that they teach.

```
((((((FAC[FNo=Fno]DEP)
        [DNo=DNo]TCH)
        [TNo=TNo]LEC)
        [SNo=SNo]SBJ)
        [FAC.Name='CSF'])
        [Post='prof'])
        [TCH.Name, SBJ.Name]
```



Examples of division operation

1) Output teacher numbers that teach in **ALL** groups:

```
((LEC[TNo,GNo])[GNo÷GNo]GRP)[TNo]
```

2) Output teacher numbers that teach in **ALL** groups of the first course:

```
((LEC[TNo,GNo])[GNo÷GNo](GRP[Course=1]))[TNo]
```

3) Output teacher names that teach in **ALL** groups of the first course:

```
(((LEC[TNo,GNo])[GNo÷GNo](GRP[Course=1]))
  [TNo=TNo]TCH)[TCH.Name]
```

GRP(GNo, Course...) **TCH(TNo, Name...)**

LEC(GNo,TNo,...)

Additional operations

Additional operations

- Assignment
- Renaming
- Generalized projection
- Outer join
- …

Assignment operation

The assignment operation (←) provides a convenient way to express complex queries, write query as a sequential program consisting of a series of assignments by using intermediate temporal relations followed by an expression whose value is displayed as a result of the query.

Пример: Output teacher names that teach in **ALL** groups of the first course:

```
(((LEC[TNo,GNo])[GNo÷GNo](GRP[Course=1]))
   [TNo=TNo]TCH)[TCH.Name]
```

```
Temp1 ← LEC[TNo,GNo]
Temp2 ← GRP[Course=1]
Temp3 ← Temp1[GNo÷GNo]Temp2
Temp4 ← Temp3[TNo=TNo]TCH
Temp4[TCH.Name]
```

Rename operation

The rename operation allows us to name, and therefore to refer to, the results of relational-algebra expressions.
It has the following syntax:

$$\rho_{R(A1,A2,...,An)}(E)$$

Where: E – relational algebra expression,
R(A1,A2,...,An) – name of the relation ant its attributes that is calculated by expression E.

Example: Output names of faculties and their departments. Rezulted relation should have name FAC_DEP with attributes FName и DName respectively:

$$\rho_{FAC\_DEP(FName,DName)}(FAC[FNo=FNo]DEP)[FAC.Name,DEP.Name]$$

Generalized projection operation

Extends the projection operation by allowing arithmetic functions to be used in the projection list.

R[F1, F2,...,Fn] $\pi_{F1,\ F2,...,Fn}(R)$

Where: R – relation or expression of relational algebra;
F1, F2,...,Fn – are arithmetic expressions involving constants and attributes in the schema of *E*.

Example: Output teacher names and their salary + commission

```
TCH[Name, Salary + Commission]
```

Outer join

Outer join is an extension of the join operation that avoids loss of information.

Computes the join and then adds tuples form one relation that does not match tuples in the other relation to the result of the join.

Outer join – example of ordinary join

FAC

FNo	Name	Dean
F-1	CSF	Ann
F-2	CTF	Dick
F-3	CEF	Bob
F-4	CYB	John

DEP

DNo	Name	Head	FNo
D-1	SE	Kate	F-1
D-2	DBMS	Lucy	F-1
D-3	CAD	Dave	F-2
D-4	PL	Stiv	NULL
D-5	CAM	Sam	NULL

1) Ordinary join
(inner join)

FAC [Fno=Fno] DEP

FAC $\bowtie$ DEP

FAC			DEP			
FNo	Name	Dean	DNo	Name	Head	FNo
F-1	CSF	Ann	D-1	SE	Kate	F-1
F-1	CSF	Ann	D-2	DBMS	Lucy	F-1
F-2	CTF	Dick	D-3	CAD	Dave	F-2

Left outer join

FAC

FNo	Name	Dean
F-1	CSF	Ann
F-2	CTF	Dick
F-3	CEF	Bob
F-4	CYB	John

DEP

DNo	Name	Head	FNo
D-1	SE	Kate	F-1
D-2	DBMS	Lucy	F-1
D-3	CAD	Dave	F-2
D-4	PL	Stiv	NULL
D-5	CAM	Sam	NULL

2) Left outer join

FAC 〈Fno=Fno] DEP

FAC ⟕ DEP

FAC / DEP

FNo	Name	Dean	DNo	Name	Head	FNo
F-1	CSF	Ann	D-1	SE	Kate	F-1
F-1	CSF	Ann	D-2	DBMS	Lucy	F-1
F-2	CTF	Dick	D-3	CAD	Dave	F-2
F-3	CEF	Bob	null	null	null	null
F-4	CYB	John	null	null	null	null

Right outer join

FAC

FNo	Name	Dean
F-1	CSF	Ann
F-2	CTF	Dick
F-3	CEF	Bob
F-4	CYB	John

DEP

DNo	Name	Head	FNo
D-1	SE	Kate	F-1
D-2	DBMS	Lucy	F-1
D-3	CAD	Dave	F-2
D-4	PL	Stiv	NULL
D-5	CAM	Sam	NULL

3) Right outer join

FAC [Fno=Fno⟩ DEP

FAC ⟖ DEP

FAC			DEP			
FNo	Name	Dean	DNo	Name	Head	FNo
F-1	CSF	Ann	D-1	SE	Kate	F-1
F-1	CSF	Ann	D-2	DBMS	Lucy	F-1
F-2	CTF	Dick	D-3	CAD	Dave	F-2
null	null	null	D-4	PL	Stiv	null
null	null	null	D-5	CAM	Sam	null

Full outer join

4) Full outer join

FAC ⟨Fno=Fno⟩ DEP

FAC ⟗ DEP

FAC			DEP			
FNo	Name	Dean	DNo	Name	Head	FNo
F-1	CSF	Ann	D-1	SE	Kate	F-1
F-1	CSF	Ann	D-2	DBMS	Lucy	
F-1						
F-2	CTF	Dick	D-3	CAD	Dave	
F-2						
F-3	CEF	Bob	null	null	null	null
F-4	CYB	John	null	null	null	null
null	null	null	D-4	PL	Stiv	null
null	null	null	D-5	CAM	Sam	null

Full outer join:
- Inner join
- Left outer join
- Right outer join

Equivalent transformations of relational expressions

1) Commutativity of selection: $\sigma_F(\sigma_G(R))=\sigma_G(\sigma_F(R))=\sigma_{F\&G}(R)$

2) Commutativity of selection and projection:

$\pi_G(\sigma_F(R))=\sigma_F(\pi_G(R))=\sigma_{F\&G}(R)$, если $G \supseteq F$

3) Distributivity of selection and product

$\sigma_F(R \times S) = \sigma_F(R) \times \sigma_F(S)$

4) Distributivity of selection and set-theoretic operations:

$\sigma_F(R \cup S)=\sigma_F(R) \cup \sigma_F(S)$, $\sigma_F(R \cap S)=\sigma_F(R) \cap \sigma_F(S)$

5) Distributivity of selection and join:

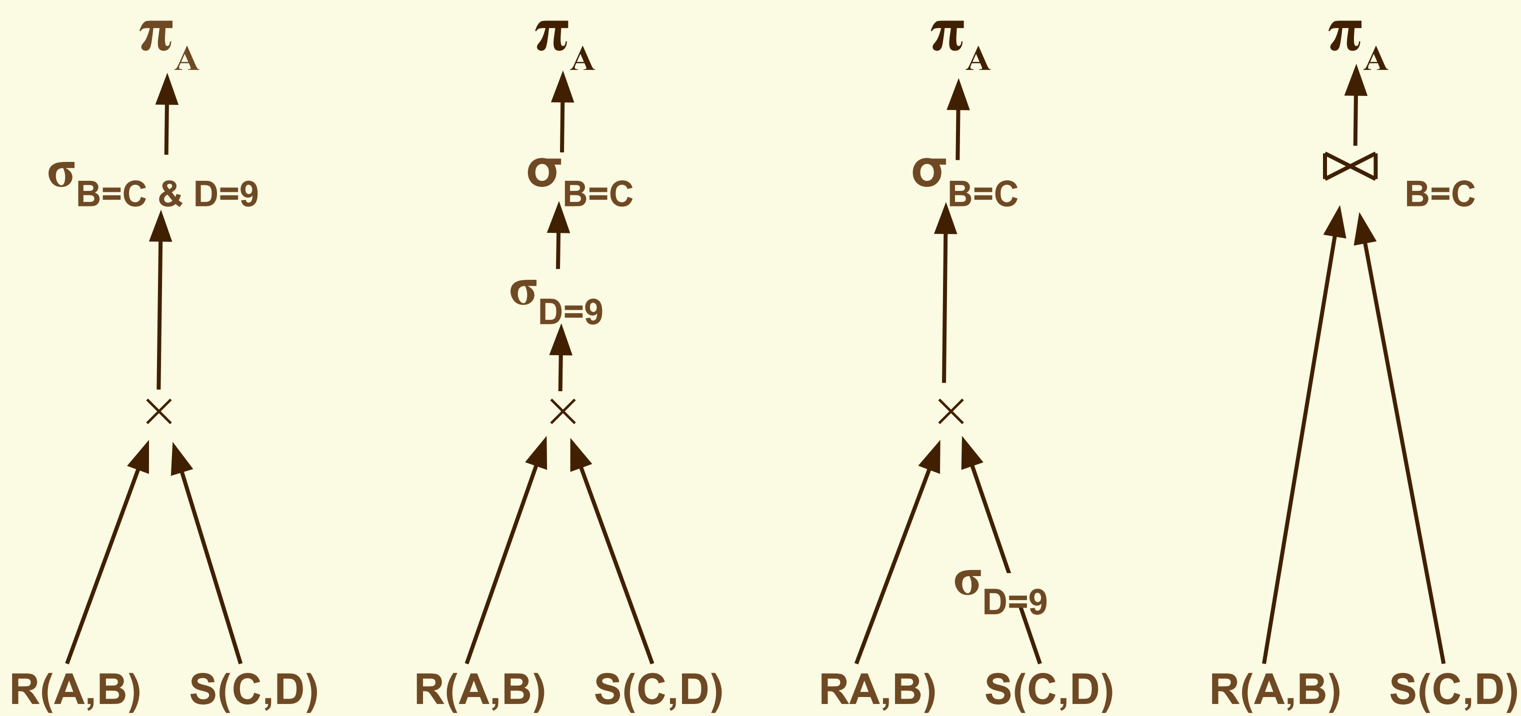
$\sigma_F(R \bowtie S) = \sigma_F(R) \bowtie S$, если условие F относится к R

6) Distributivity of projection and set-theoretic operations :

$\pi_F(R \cup S)=\pi_F(R) \cup \pi_F(S)$, $\pi_F(R \cap S)=\pi_F(R) \cap \pi_F(S)$

Optimization of RA expressions

$$\pi_A(\sigma_{B=C\ \&\ D=9}(R(A, B) \times S(C,D)))$$



General rules of RA expressions optimization

General rules of RA expressions optimization:

- Transform each selection $\sigma_{F1\&...\&Fn}(E)$ to the sequence of selections $\sigma_{F1}(... \sigma_{Fn}(E))$
- Move each selection downwards of the tree as far as it is possible (thus (vertical) size of the relation is reduced).
- Adjacent selection and cartesian product are replaced by join.
- Move each projection downwards of the tree as far as it is possible (thus (horizontal) size of the relation is reduced).
- Transform each cascade of adjacent selections and projections into single projection or selection with subsequent projection

Relational Algebra: Summary

- Relational Algebra:
  - Formal language for handling data in relational model
  - Procedural language, *how to* retrieve data
  - No practical relevance for querying DB
  - Formal basis for query optimization
- Important terms & concepts:
  - Union $R \cup S$, difference $R - S$, intersection $R \cap S$
  - Projection $\pi_{\text{<attribute list>}}(R)$
  - Selection $\sigma_{\text{<predicate>}}(R)$
  - Cartesian product $R \times S$
  - Joins $R \bowtie_{\text{<predicate>}} S$
  - Difference