Le modèle relationnel

Patricia Serrano Alvarado

Master Bio Informatique

UNIVERSITÉ DE NANTES

Merci à Emmanuel Desmontils

Plan

- Structure de données
  - Terminologie (relation, attribut, cardinalité, degré, tuple, domaine, clé primaire, etc.)
- Manipulation de données
  - L'algèbre relationnelle
- Intégrité de données
  - Contraintes, clés et déclencheurs

Introduction

- SGBD relationnelles :
  - Données en tables (ou relations)
  - Requêtes exprimées à l'aide d'opérateurs sur ces tables
- Avantages
  - Pour un ensemble d'informations à trouver, on décrit juste les opérateurs sur les relations qui permettent de les atteindre (expressions relationnelles)
  - Approche beaucoup plus détachée de l'implémentation physique
  - Simple, facile à comprendre, logique et fidèle à un cadre mathématique (algèbre relationnelle, théorie des ensembles)

Domaine et tuple

- Domaine : ensemble fini ou infini de valeurs distinctes que peut prendre une donnée élémentaire
  - Entiers, Booléens...
- n-uplet, tuple : élément $<v_1,v_2 \dots v_n>$ du produit cartésien d'un ensemble de domaines $D_1, D_2 \dots D_n$ noté $D_1 \times D_2 \times \dots D_n$ tel que $\forall i, v_i \in D_i$.
  - Exemple : si $D_1=\{a,b,c\}$ et $D_2=\{1,2\}$ alors $D_1 \times D_2 = \{<a,1>, <a,2>, <b,1>, <b,2>, <c,1>, <c,2>\}$
- Remarque : le produit cartésien peut être généralisé sur des ensembles quelconques et, en particulier, des ensembles de tuples.

Exemples de domaines

- Des domaines
  - Nom = {Dupont, Martin, Robert, Denou, Bellamy...}
  - Prénom = {Franck, Isabelle, Adrien, Michelle, Jean, Gustave...}
  - Note = [0...20]
  - ...
- Des tuples :
  - Nom ∢ Prénom = {<Dupont,Franck>, <Dupont,Isabelle>..., <Martin,Franck>, <Martin,Isabelle>...}
  - Nom ∢ Note = {<Dupont,0>, <Dupont,1>...}
  - ...

Tables/Relations

- Une table ou relation : sous-ensemble du produit cartésien d'un ensemble de domaines. Elle est identifiée par un nom.
- Représentation : tableau dont les lignes sont les tuples. Les noms de colonnes sont les attributs de la table. Chaque attribut est unique.
  - L'ordre des lignes et des colonnes est sans importance.
  - Tous les tuples doivent être distincts : Notion de clé
- Le degré d'une relation R : nombre d'attributs de R.
  - Notée : $\delta(R)$
    - Nombre de colonnes du tableau
- La cardinalité d'une relation R : nombre de tuples.
  - Notée : Card(R) ou |R|
    - Nombre de lignes du tableau

Schéma

- Schéma d'une relation
  - Le nom de la table
  - L'ensemble des noms des attributs
  - Les contraintes...

- Schéma d'une base de données : ensemble des schémas de toutes les relations.

Base de données relationnelle

- Base de données relationnelle : base de données dont le schéma est un ensemble de schémas de tables relationnelles et dont les occurrences sont des tuples de ces tables.

- Expression relationnelle : expression de syntaxe précise composée de noms de relations, d'attributs, de constantes et d'opérations. Elle décrit une relation.

- Langage relationnel : décrit des expressions relationnelles.

Exemple de base de données

δ(Notes) = 4
|Notes| = 11

Notes	noe	codemat	noteex	notecc
	99628C	MIAS2I5	12	15,5
	46283B	MIAS2I6	8	11
	46283B	MIAS2I5	9,5	2
	86719E	IUP2MA	12	5,5
	99321C	LIL6	18	16,5
	28936E	MIAS2I5	13,5	13,5
	86719E	IUP2IS	8,5	10
	99628C	MIAS2I6	3	6
	99321C	LIL5	15	14,5
	99322C	MIAS2I5	12	15,5
	28936E	MIAS2I6	12	null

δ(Étudiants) = 3
|Étudiants| = 6

Étudiants	noetu	nom	prénom
	28936E	Dupont	Franck
	46283B	Dupont	Isabelle
	86719E	Martin	Adrien
	99628C	Robert	Adrien
	99321C	Denou	Michelle
	99322C	Dupont	Isabelle

δ(Matières) = 4
|Matières| = 7

Matières	codemat	titre	responsable	diplôme
	MIAS2I5	I5	E238	Deug MIAS
	MIAS2I6	I6	E426	Deug MIAS
	IUP2MA	Automates	E238	Licence IUP-MIAGE
	LIL6	Systèmes	E236	Licence Informatique
	IUP2IS	Systèmes	E526	Licence IUP-MIAGE
	MIAS2I3	Math-Info	E426	Deug MIAS
	LIL5	Algo	E426	Licence Informatique

Exemple de requête

- Tous les étudiants ayant leur moyenne à l'examen de 15

```
Select noetu, nom, prenom
From Etudiants, Notes, Matieres
Where Etudiants.noetu=Notes.noe And
Matiere.codemat=Notes.codemat And noteex>=10
And titre='I5';
```

Clé d'une relation

- L'ensemble d'attributs permettant d'identifier de manière unique un tuple
  - Permet de différencier chacun des tuples d'une table
  - De préférence, l'ensemble est le plus petit possible

Exemple de base de données

Notes

noe	codemat	noteex	notecc
99628C	MIAS2I5	12	15,5
46283B	MIAS2I6	8	11
46283B	MIAS2I5	9,5	2
86719E	IUP2MA	12	5,5
99321C	LIL6	18	16,5
28936E	MIAS2I5	13,5	13,5
86719E	IUP2IS	8,5	10
99628C	MIAS2I6	3	6
99321C	LIL5	15	14,5
99322C	MIAS2I5	12	15,5
28936E	MIAS2I6	12	null

Étudiants

noetu	nom	prénom
28936E	Dupont	Franck
46283B	Dupont	Isabelle
86719E	Martin	Adrien
99628C	Robert	Adrien
99321C	Denou	Michelle
99322C	Dupont	Isabelle

Matières

codemat	titre	responsable	diplôme
MIAS2I5	I5	E238	Deug MIAS
MIAS2I6	I6	E426	Deug MIAS
IUP2MA	Automates	E238	Licence IUP-MIAGE
LIL6	Systèmes	E236	Licence Informatique
IUP2IS	Systèmes	E526	Licence IUP-MIAGE
MIAS2I3	Math-Info	E426	Deug MIAS
LIL5	Algo	E426	Licence Informatique

Algèbre rationnelle

Algèbre rationnelle et SQL

- Algèbre relationnelle : Codd 1970
  - Formaliser les opérations sur les ensembles.
  - Opérations de base : ensemble minimal d'opérations telles que toute autre opération peut être déduite de celles-ci.
    - Unaires (2) ou binaires (3)
  - Opérations dérivées : opérations définies à partir des 5 opérateurs de base.
- SQL : langage relationnel extrêmement répandu
  - Très grande majorité des SGBD-R utilisent ce langage
  - Select ... From ... Where ... Group By ... Having ... Order By ... ;
  - Create Table ... (...) ;
  - ...

Opérations de bases unaires

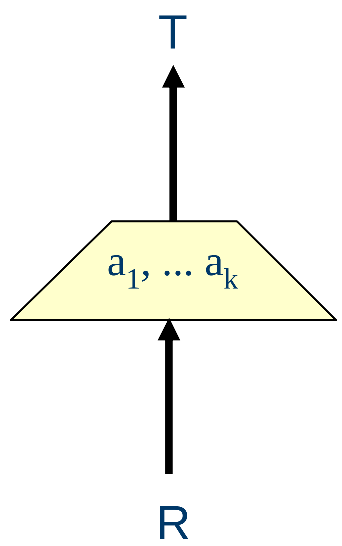
- Opérations de base de l'algèbre relationnelle opérant sur une seule table.
- Objectif : éliminer des attributs ou des tuples dans une relation

- Projection : réduire le nombre d'attributs pour ne garder que ceux considérés comme "utiles".

- Sélection : réduire le nombre de tuples pour ne garder que ceux considérés comme significatifs.

La projection

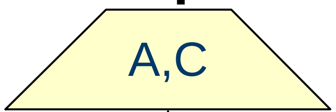
Arguments	$R(A_1, A_2, \ldots A_n)$
Notation	$T= \prod_{a1,\ldots ak}(R)$ = Project($R, a_1, \ldots a_k$) = $R(a_1, \ldots a_k)$
Schéma	$T(a_1, \ldots a_k)$
Valeur	t-uples de R dont on ne garde que les attributs $a_1, \ldots a_k$. Les doublons sont supprimés.
SQL	Select distinct $a_1, \ldots a_k$ From R;
Remarque	$\delta(\prod(R))\leq\delta(R)$ et $\lvert\prod(R)\rvert\leq\lvert R\rvert$



Exemple de projection

$\Pi_{A,C}(R_1)$

A	C
a	c
d	f
c	d



R_1

A	B	C
a	b	c
d	a	f
c	b	d
a	e	c

Select Distinct A,C From R_1;

R_1

A	B	C
a	b	c
d	a	f
c	b	d
~~a~~	e	~~c~~

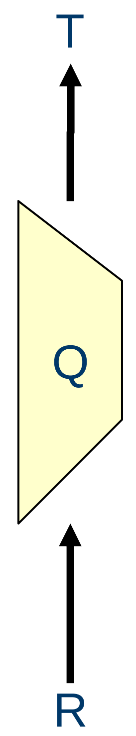
Clé d'une relation

- Tous les tuples de la relation se distinguent deux à deux par leur projection selon la clé.
- D'où : $|R| = |\prod_{Clé}(R)|$

Étudiants	noetu	nom	prénom
	28936E	Dupont	Franck
	46283B	Dupont	Isabelle
	86719E	Martin	Adrien
	99628C	Robert	Adrien
	99321C	Denou	Michelle
	99322C	Dupont	Isabelle

La sélection

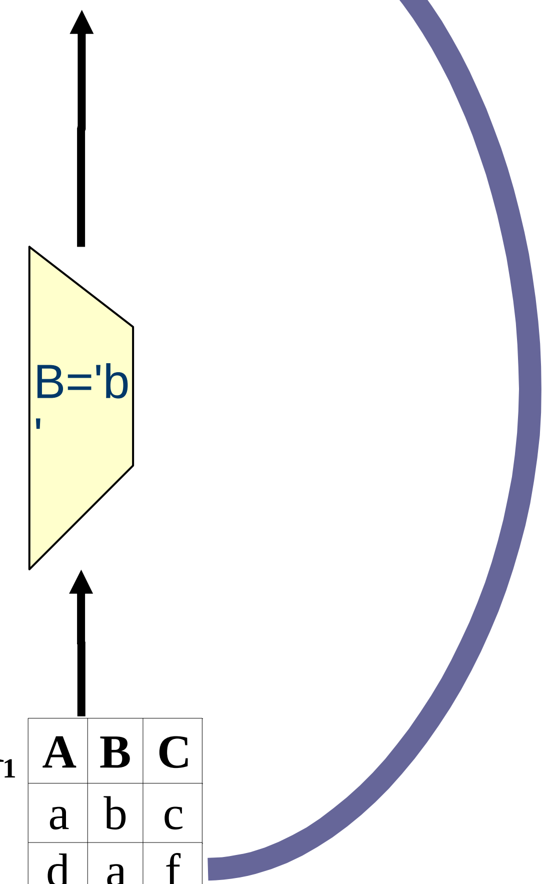
Arguments	$R(A_1, A_2, \ldots A_n)$
Notation	$T= \sigma_Q(R) = Cond_Q(R) = R(Q) =$ $Select(R,Q) = Restrict(R,Q)$ avec Q une formule logique sur les attributs (négation, conjonction ou disjonction entre termes des la forme $A_i \theta A_j$ ou $A_i \theta a$ ou A_i et A_j sont des attributs, a une constante et θ un opérateur de comparaison)
Schéma	$T(A_1, A_2, \ldots A_n)$
Valeur	t-uples de R satisfaisant la qualification Q.
SQL	Select * From R Where Q;
Remarque	$\delta(\sigma(R))=\delta(R)$ et $\|\sigma(R)\|\leq\|R\|$



Exemple de sélection

$\sigma_{B='b'}(R_1)$

A	B	C
a	b	c
c	b	d



R_1

A	B	C
a	b	c
d	a	f
c	b	d
a	e	c

Select * From R_1 Where B='b';

R_1

A	B	C
a	b	c
d	a	f
c	b	d
a	e	c

Opérations de base ensemblistes

- Combinaison de deux relations pour en former une troisième.

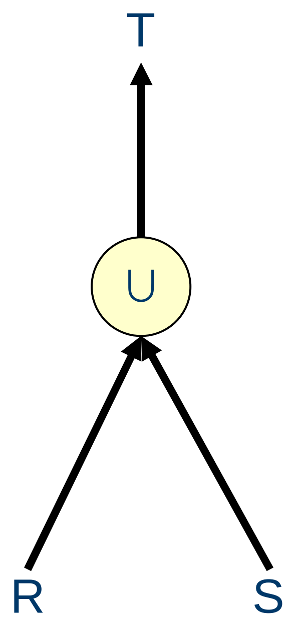
- Union : regroupement de données similaires.

- Différence : réduction d'un ensemble de données.

- Produit cartésien : peu utilisé directement (combinaison de données).

L'union

Arguments	$R(A_1, A_2, \ldots A_n)$ $S(A_1, A_2, \ldots A_n)$
Notation	$T = R \cup S$ = Union(R,S) = Or(R,S) = Append(R,S)
Schéma	$T(A_1, A_2, \ldots A_n)$
Valeur	Union ensembliste sur $D_1 \times D_2 \times \ldots D_n$ telle que $T=\{t \mid t \in R \vee t \in S\}$
SQL	Select * From R Union Select * From S;
Remarque	$\delta(R \cup S) = \delta(R) = \delta(S)$ et $\|R \cup S\| \leq R\| + \|S\|$



Exemple d'union

$R_1 \cup S_1$

A	B	C
a	b	c
d	a	f
c	b	d
a	e	c
b	g	a

∪

R_1

A	B	C
a	b	c
d	a	f
c	b	d
a	e	c

S_1

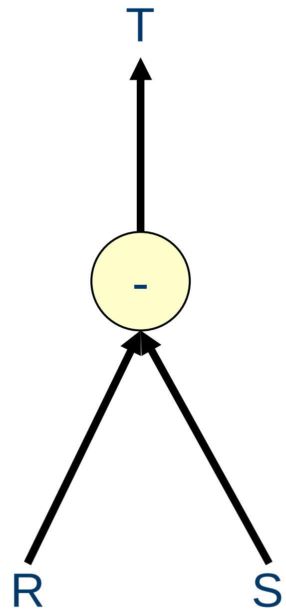
A	B	C
b	g	a
d	a	f

Select * From R_1
Union
Select * From S_1 ;

	A	B	C
R_1	a	b	c
	d	a	f
	c	b	d
	a	e	c
S_1	b	g	a
	d	a	f

La différence

Arguments	$R(A_1, A_2, \ldots A_n)$ $S(A_1, A_2, \ldots A_n)$
Notation	T= R - S = Diff(R,S) = Remove(R,S) = Minus(R,S)
Schéma	$T(A_1, A_2, \ldots A_n)$
Valeur	Différence ensembliste sur $D_1 \times D_2 \times \ldots D_n$ telle que $T=\{t \mid t \in R \wedge t \notin S\}$
SQL	Select * From R Except Select * From S;
Remarque	$\delta(R-S)=\delta(R)=\delta(S)$ et $\lvert R-S \rvert \leq \lvert R \rvert$



Exemple de différence

R_1-S_1

A	B	C
a	b	c
c	b	d
a	e	c

-

R_1

A	B	C
a	b	c
d	a	f
c	b	d
a	e	c

S_1

A	B	C
b	g	a
d	a	f

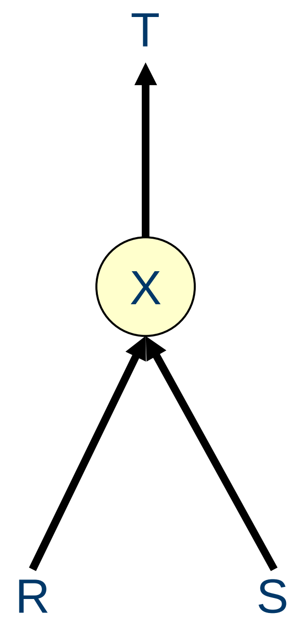
Select * From R_1
Except
Select * From S_1 ;

R_1

A	B	C
a	b	c
d	a	f
c	b	d
a	e	c

Le produit cartésien

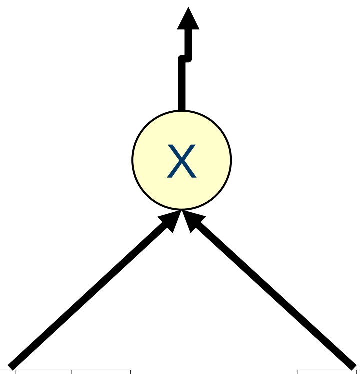
Arguments	$R(A_1, A_2, \ldots A_n)$ $S(B_1, B_2, \ldots B_m)$
Notation	T= R × S = Prod(R,S) = Product(R,S) = Times(R,S)
Schéma	$T(A_1, A_2, \ldots A_n, B_1, B_2, \ldots B_m)$
Valeur	Ensemble de tous les t-uples ayant n+m attributs dont les n premiers forment un t-uple de R et les m derniers un t-uples de S
SQL	Select * From R , S; Select * From R Cross Join S;
Remarque	$\delta(R\times S)=\delta(R)+\delta(S)$ et $\lvert R\times S\rvert=\lvert R\rvert*\lvert S\rvert$



Exemple de produit cartésien

$R_1 \times S_1$

$R_1.A$	$R_1.B$	$R_1.C$	$S_1.A$	$S_1.B$	$S_1.C$
a	b	c	b	g	a
a	b	c	d	a	f
d	a	f	b	g	a
d	a	f	d	a	f
c	b	d	b	g	a
c	b	d	d	a	f
a	e	c	b	g	a
a	e	c	d	a	f



R_1

A	**B**	**C**
a	b	c
d	a	f
c	b	d
a	e	c

S_1

A	**B**	**C**
b	g	a
d	a	f

Select * From R_1, S_1 ;

$R_1 \times S_1$

$R_1.A$	$R_1.B$	$R_1.C$	$S_1.A$	$S_1.B$	$S_1.C$
a	b	c	b	g	a
a	b	c	d	a	f
d	a	f	b	g	a
d	a	f	d	a	f
c	b	d	b	g	a
c	b	d	d	a	f
a	e	c	b	g	a
a	e	c	d	a	f

Opérations dérivées

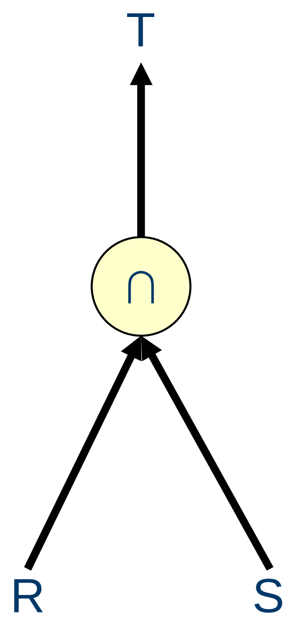
- Intersection

- Complément

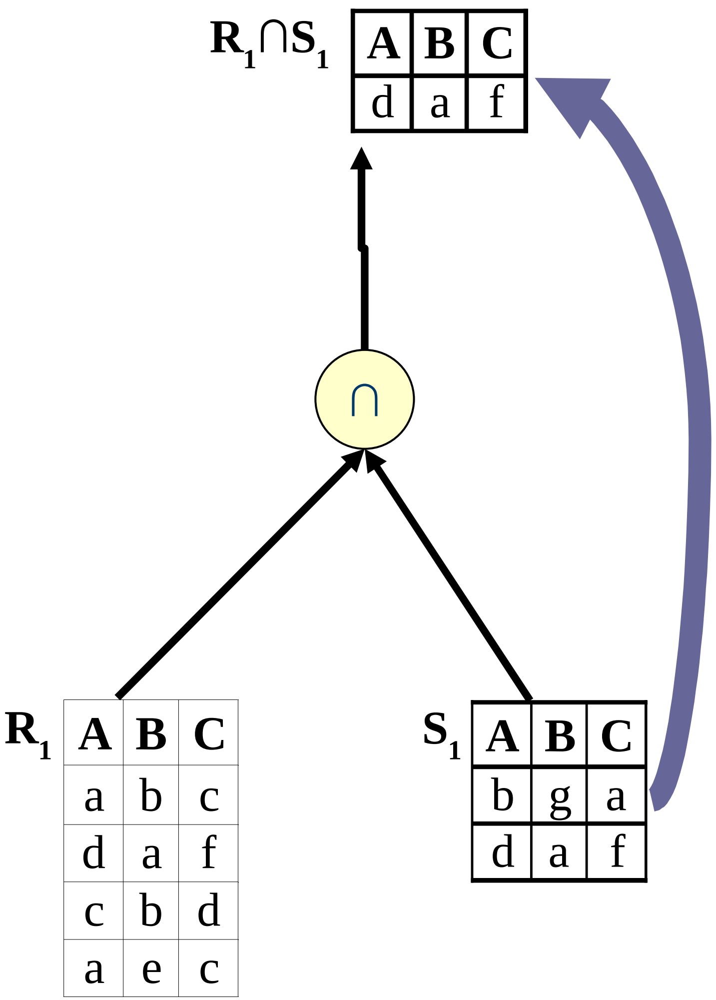
- θ-jointure : opération majeure dans les SGBDR
  - Elle permet d'enrichir des tuples et de mettre en relation des attributs.
  - Jointure naturelle
  - Semi-jointure
  - Jointure externe

L'intersection

Arguments	$R(A_1, A_2, \ldots A_n)$ $S(A_1, A_2, \ldots A_n)$
Notation	$T= R \cap S = \text{Inter}(R,S) = \text{And}(R,S)$
Schéma	$T(A_1, A_2, \ldots A_n)$
Valeur	Intersection ensembliste sur $D_1 \times D_2 \times \ldots D_n$ telle que $T=\{t \mid t \in R \wedge t \in S\} = R-(R-S) = S-(S-R)$
SQL	Select * From R Intersect Select * From S;
Remarque	$\delta(R \cap S)=\delta(R)=\delta(S)$ et $\lvert R \cap S \rvert \leq \lvert R \rvert$ et $\lvert R \cap S \rvert \leq \lvert S \rvert$



Exemple d'intersection



Select * From R_1
Intersect
Select * From S_1 ;

R_1

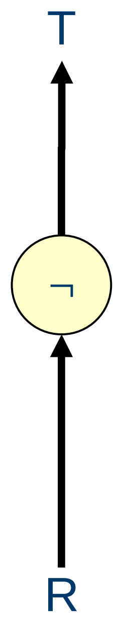
A	B	C
a	b	c
d	a	f
c	b	d
a	e	c

S_1

A	B	C
b	g	a
d	a	f

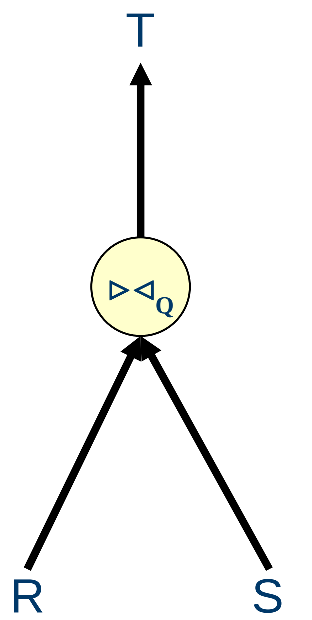
Le complément

Arguments	$R(A_1, A_2, \ldots A_n)$
Notation	$T = \neg R = Comp(R)$
Schéma	$T(A_1, A_2, \ldots A_n)$
Valeur	Complémentaire ensembliste sur $D_1 \times D_2 \times \ldots D_n$ telle que $T=\{t \mid t \notin R\} = (D_1 \times D_2 \times \ldots D_n) - R$
SQL	
Remarque	$\delta(Comp(R)) = \delta(R)$



La jointure interne

$R(A_1, A_2, \ldots A_n)$ $S(B_1, B_2, \ldots B_m)$	
$T = R \bowtie_Q S$ = Join(R,S,Q) = $Join_Q(R,S)$ avec Q une combinaison logique de termes de la forme $A_i Op B_j$	
$T(A_1, A_2, \ldots A_n, B_1, B_2, \ldots B_m)$	
t-uples de R× S vérifiant la qualification Q. $T = R \bowtie_Q S = \sigma_Q(R \times S)$	
Select * From R Inner Join S On Q; Select * From R Join S On Q;	
$\delta(R \bowtie_Q S) = \delta(R) + \delta(S)$ et $\|R \bowtie_Q S\| \leq \|R\| * \|S\|$ $R \bowtie_Q R$: auto-jointure	



Exemple de jointure

$R_3 \bowtie_{B<D} S_3$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	9
4	5	6	6	9

$\bowtie_{B<D}$

R_3

A	B	C
1	2	3
4	5	6
7	8	9

S_3

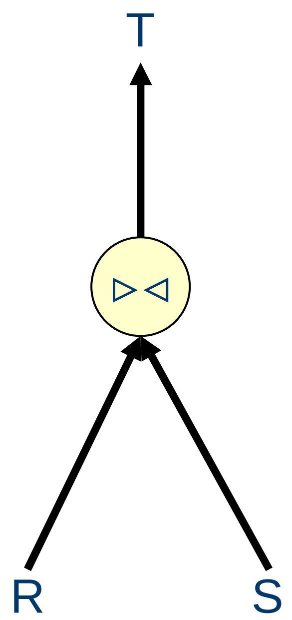
D	E
3	1
6	9

Select * From R_3 Join S_3 On B<D;

A	B	C	D	E
1	2	3	3	1
1	2	3	6	9
4	5	6	3	1
4	5	6	6	9
7	8	9	3	1
7	8	9	6	9

La jointure naturelle

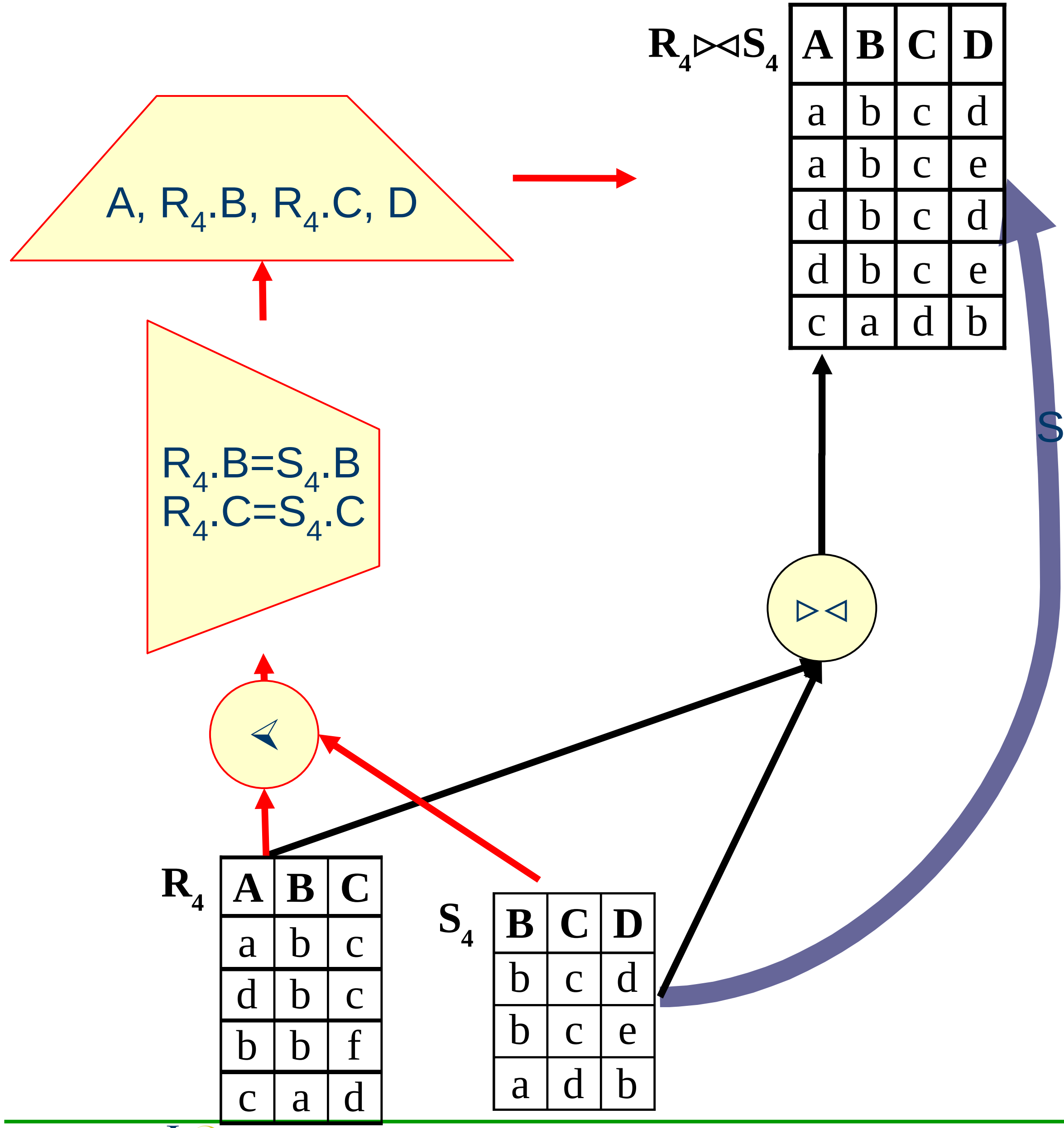
Arguments	$R(X_1, X_2, \ldots X_p, A_1, A_2, \ldots A_n)$ $S(X_1, X_2, \ldots X_p, B_1, B_2, \ldots B_m)$
Notation	$T = R \bowtie S$
Schéma	$T(X_1, X_2, \ldots X_p, A_1, A_2, \ldots A_n, B_1, B_2, \ldots B_m)$
Valeur	Ensemble de tous les t-uples ayant m+n+p attributs dont les p+n premiers forment un t-uple de R et les p premiers et les m derniers un t-uple de S.
SQL	Select * From R Natural Join S On Q;
Remarque	C'est la jointure la plus utilisée.



$$R \quad S = \Pi_{X_1, \ldots X_p, A_1, \ldots A_n, B_1, \ldots B_m}(\sigma_{\forall i \in [1..p], R.X_i = S.X_i}(R \times S))$$

$$R \quad S \neq R \quad S_{\forall i \in [1..p], R.X_i = S.X_i}$$

Exemple de jointure naturelle



$R_4 \bowtie S_4$

A	B	C	D
a	b	c	d
a	b	c	e
d	b	c	d
d	b	c	e
c	a	d	b

R_4

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

S_4

B	C	D
b	c	d
b	c	e
a	d	b

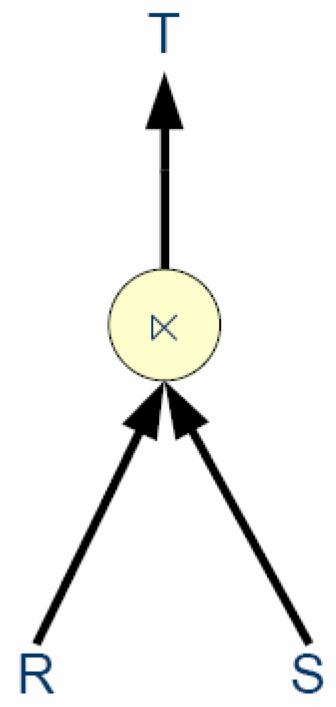
Select * From R_4 Natural Join S_4 ;

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

B	C	D
b	c	d
b	c	e
a	d	b

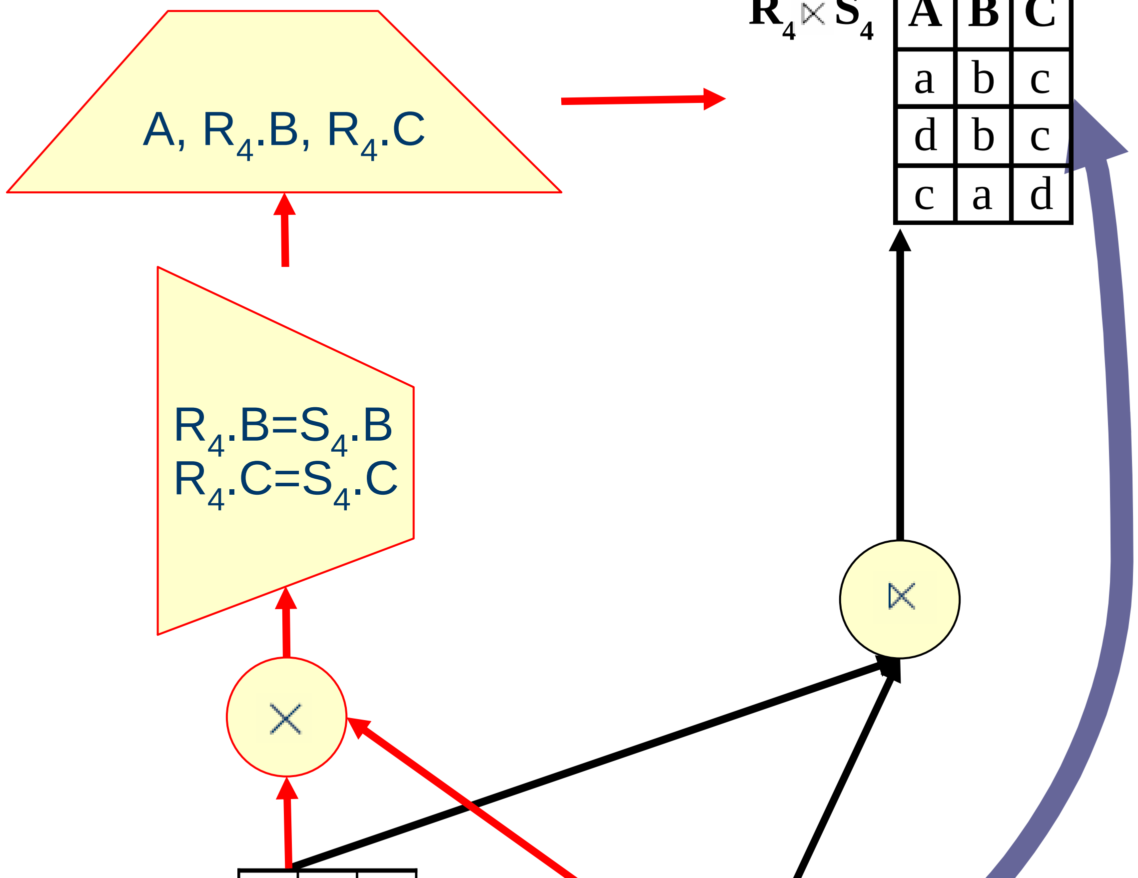
La semi-jointure

Arguments	$R(X_1, X_2, \dots X_p, A_1, A_2, \dots A_n)$ $S(X_1, X_2, \dots X_p, B_1, B_2, \dots B_m)$
Notation	$T = R \ltimes S$
Schéma	$T(X_1, X_2, \dots X_p, A_1, A_2, \dots A_n)$
Valeur	Projection sur les attributs de R de la jointure naturelle de R avec S
SQL	
Remarque	



$$R \quad S = \Pi_{X_1, \dots X_p, A_1, \dots A_n}(\sigma_{\forall i \in [1..p], R.X_i = S.X_i}(R \times S))$$
$$R \quad S = \Pi_{X_1, \dots X_p, A_1, \dots A_n}(R \quad S)$$

Exemple de semi-jointure



$R_4 \ltimes S_4$

A	B	C
a	b	c
d	b	c
c	a	d

R_4

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

S_4

B	C	D
b	c	d
b	c	e
a	d	b

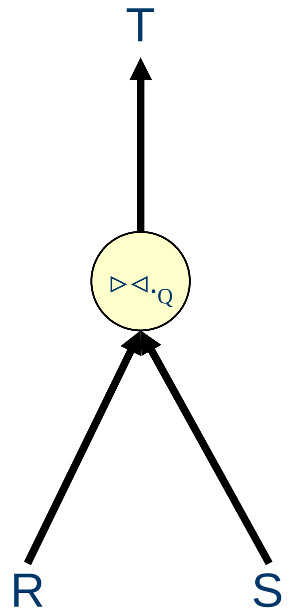
Select Distinct A,B,C
From R_4 Natural Join S_4 ;

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

B	C	D
b	c	d
b	c	e
a	d	b

La jointure externe

Arguments	$R(A_1, ..., A_n)$ $S(B_1, ..., B_m)$
Notation	$T = R \dot{\bowtie}_Q S$; $T = R\cdot \bowtie_Q S$; $T = R \bowtie \cdot_Q S$ avec Q une combinaison logique $(\wedge, \vee, \neg)$ de termes de la forme $A_i Op B_j$ et $Op \in \{=, <, >, \leq, \geq, \neq\}$
Schéma	$T(A_1, ..., A_n, B_1, ..., B_m)$
Valeur	t-uples de $R \times S$ vérifiant la qualification Q. + Ajout les t-uples d'une relation (dite directrice) ne vérifiant pas la condition en complétant les attributs de l'autre relation par des "null" (vide...). Les deux relations peuvent être directrices (position du point).
SQL	Select * From R Left Outer Join S On Q; Select * From R Right Outer Join S On Q; Select * From R Full Outer Join S On Q; Select * From R,S Where ...; avec $\oplus$ du côté des attributs à compléter



Exemple de jointure externe

$R_4 \,^{\bullet}\!\bowtie \ldots S_4$

A	B	C	B	C	D
a	b	c	b	c	d
a	b	c	b	c	e
d	b	c	b	c	d
d	b	c	b	c	e
b	b	f	b	f	null
c	a	d	a	d	b

$^{\bullet}\!\bowtie_{R4.B=S4.B,\ R4.C=S4.C}$

R_4

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

S_4

B	C	D
b	c	d
b	c	e
a	d	b

Select * From R_4 Left Outer Join S_4
On R_4.B=S_4.B and R_4.C=S_4.C;

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

B	C	D
b	c	d
b	c	e
Null	Null	Null
a	d	b

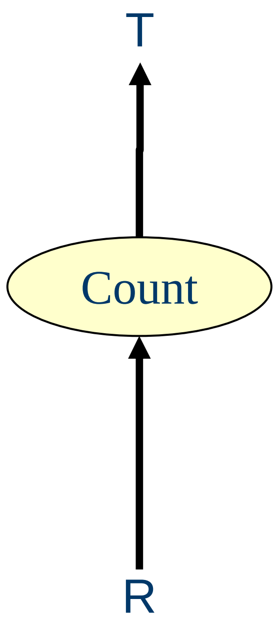
Select * From R_4 , S_4
Where ⊕R_4.B=S_4.B and ⊕R_4.C=S_4.C;

Opérations de comptage

- Opérations "outils" qui sont des extensions des opérateurs de base et non fonction de ces opérateurs.
  - Reste cependant des opérateurs ensemblistes puisqu'ils prennent des ensembles en entrée et retournent un ensemble.
- Comptage
- Somme
- Moyenne
- Minimum
- Maximum...

Comptage

Arguments	$R(A_1, ..., A_n)$
Notation	$T = Compte_{X_1,...,X_p}(R) = Count_{X_1,...,X_p}(R)$ ou $T = Compte(R)$ avec $\forall i \in [1..p], X_i \in \{A_1, ..., A_n\}$
Schéma	$T(X_1, ..., X_p, Cpt)$ ou $T(Cpt)$
Valeur	Dénombrer les t-uples qui ont une même valeur d'attributs en commun. Si les attributs ne sont pas spécifiés, c'est le nombre de t-uples.
SQL	Select Count(*) From R [Group By ...];
NB	$\delta(Count(R)) = 1$, $\delta(Count_{X_1,...,X_p}(R)) = p + 1$ $\|Count(R)\| = 1$ et $\|Count_{X_{[1..p]}}(R)\| = \|\Pi_{X_{[1..p]}}(R)\|$



Exemples de comptage

Count$_B$(T)

B	Count
n	2
m	2
o	1
p	1

Count(T)

Count
6

Count(T)

Count$_B$(T)

T

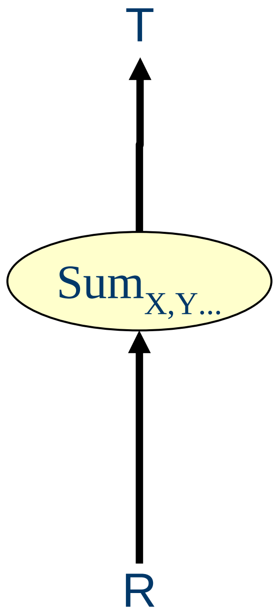
A	B	C
a	n	17
b	o	14
c	n	9
d	p	13
e	m	20
f	m	10

Select Count(*) From T;

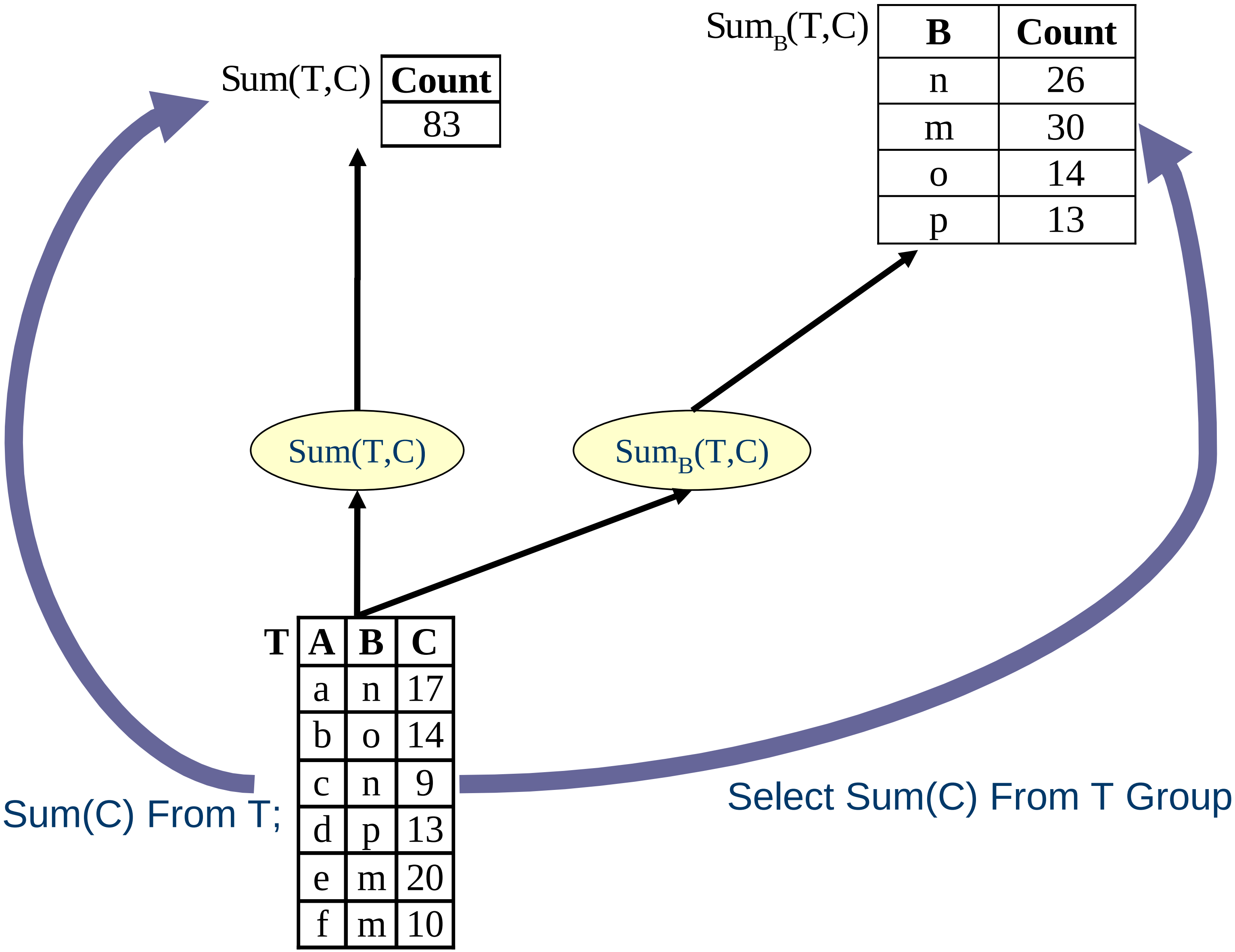
Select Count(*) From T Group By B;

Somme

Arguments	$R(A_1, ..., A_n)$
Notation	$T = Somme_{X_1,...,X_p}(R, Y) = Sum_{X_1,...,X_p}(R, Y)$ ou $T = Somme(R, Y)$ avec $\forall i \in [1..p], X_i \in \{A_1, ..., A_n\}$ et $Y \in \{A_1, ..., A_n\}$
Schéma	$T(X_1, ..., X_p, Cpt)$ ou $T(Cpt)$
Valeur	Somme cumulée des valeurs d'un attribu Y pour chacune des valeurs différentes des attributs. Si les attributs ne sont pas spécifiés, c'est la somme des Y pout tout les t-uples de R.
SQL	Select Sum(...) From R Group By ...;
NB	On peut généraliser au calculs de la moyenne, du minimum, du maximum...



Exemples de sommes



Sum(T,C)

Count
83

Sum$_B$(T,C)

B	Count
n	26
m	30
o	14
p	13

T

A	B	C
a	n	17
b	o	14
c	n	9
d	p	13
e	m	20
f	m	10

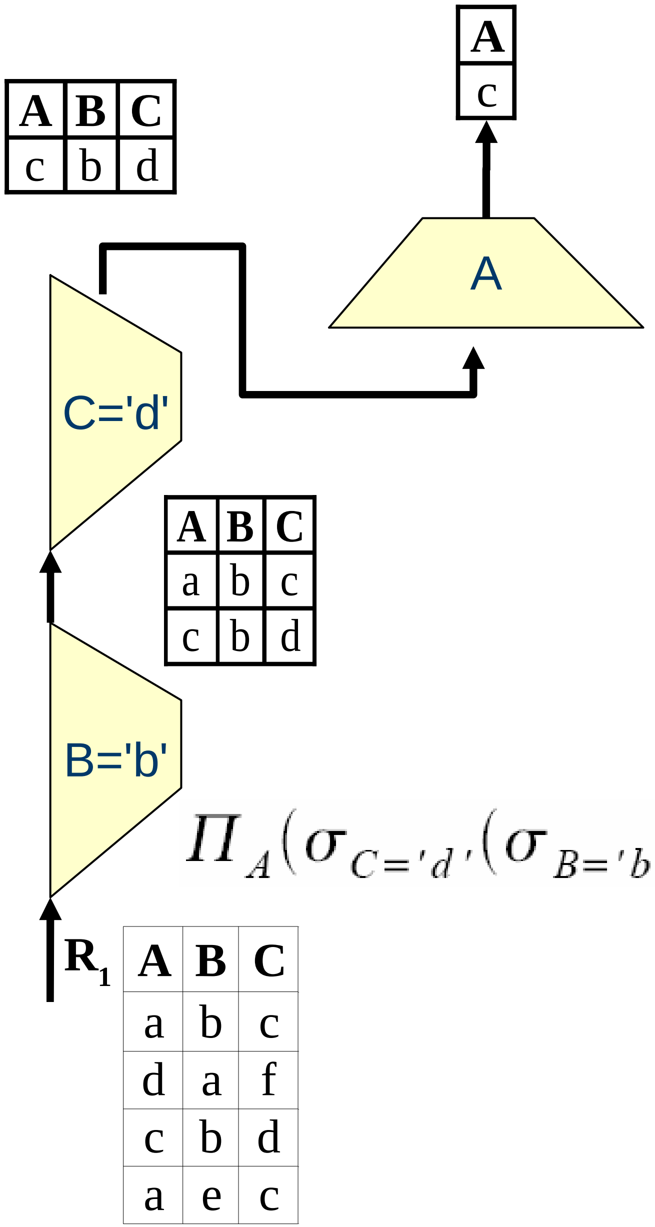
Select Sum(C) From T;

Select Sum(C) From T Group By B;

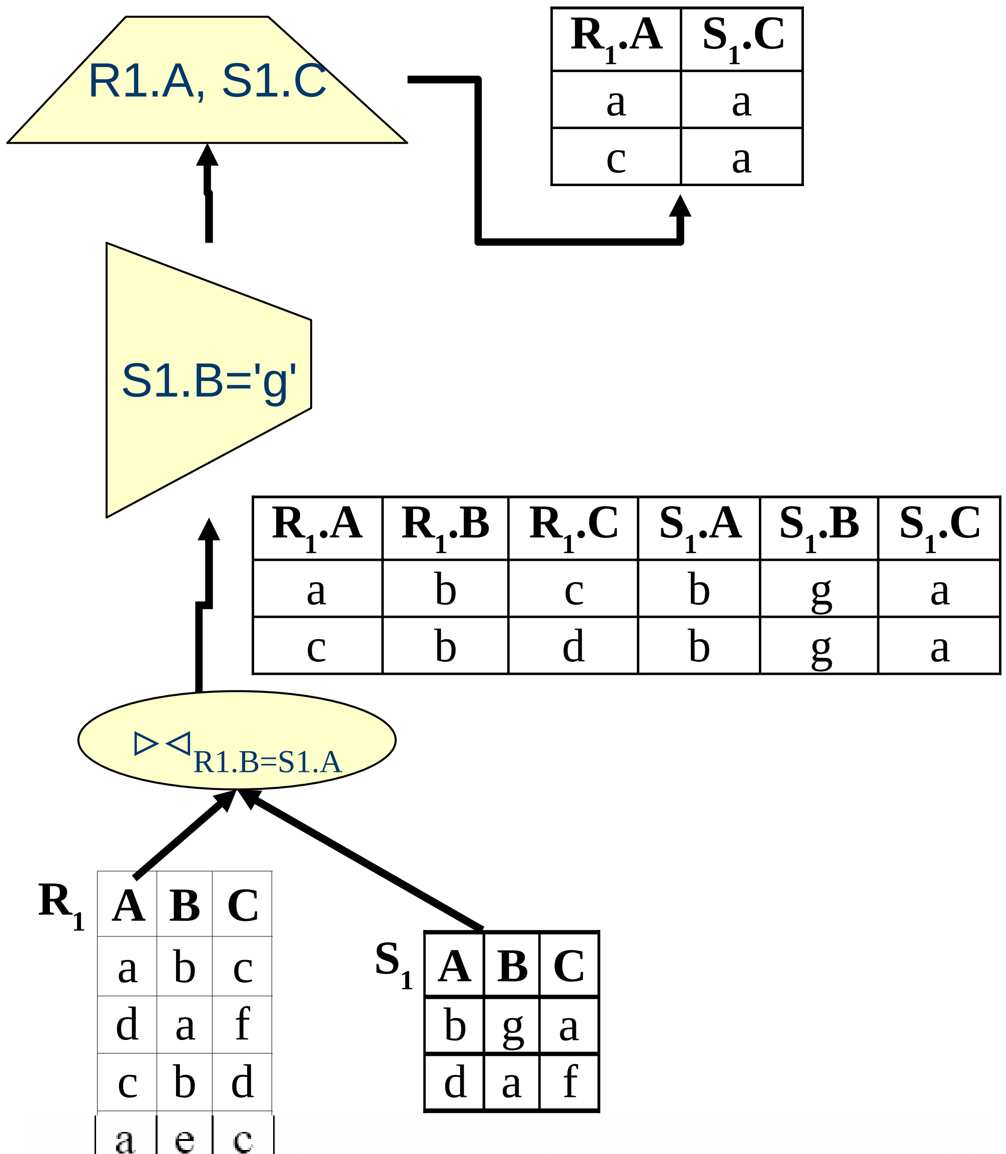
Expression

- Une expression de l'algèbre relationnelle est une expression composée de relations et d'opérateurs de l'algèbre relationnelle (de base ou non).*
  - Une expression peut être représentée à l'aide d'un arbre syntaxique.
- Le résultat d'une expression est une relation sur laquelle on peut faire une autre opération.
- Remarque : en SQL, une table peut être une donnée de base ou le résultat d'une expression.

Exemples d'expressions



$$\Pi_A(\sigma_{C='d'}(\sigma_{B='b'}(R_1)))$$



$$\Pi_{R_1.A, S_1.C}(\sigma_{S_1.B='g'}(R_1 \bowtie_{R_1.B=S_1.A} S_1))$$

Propriétés

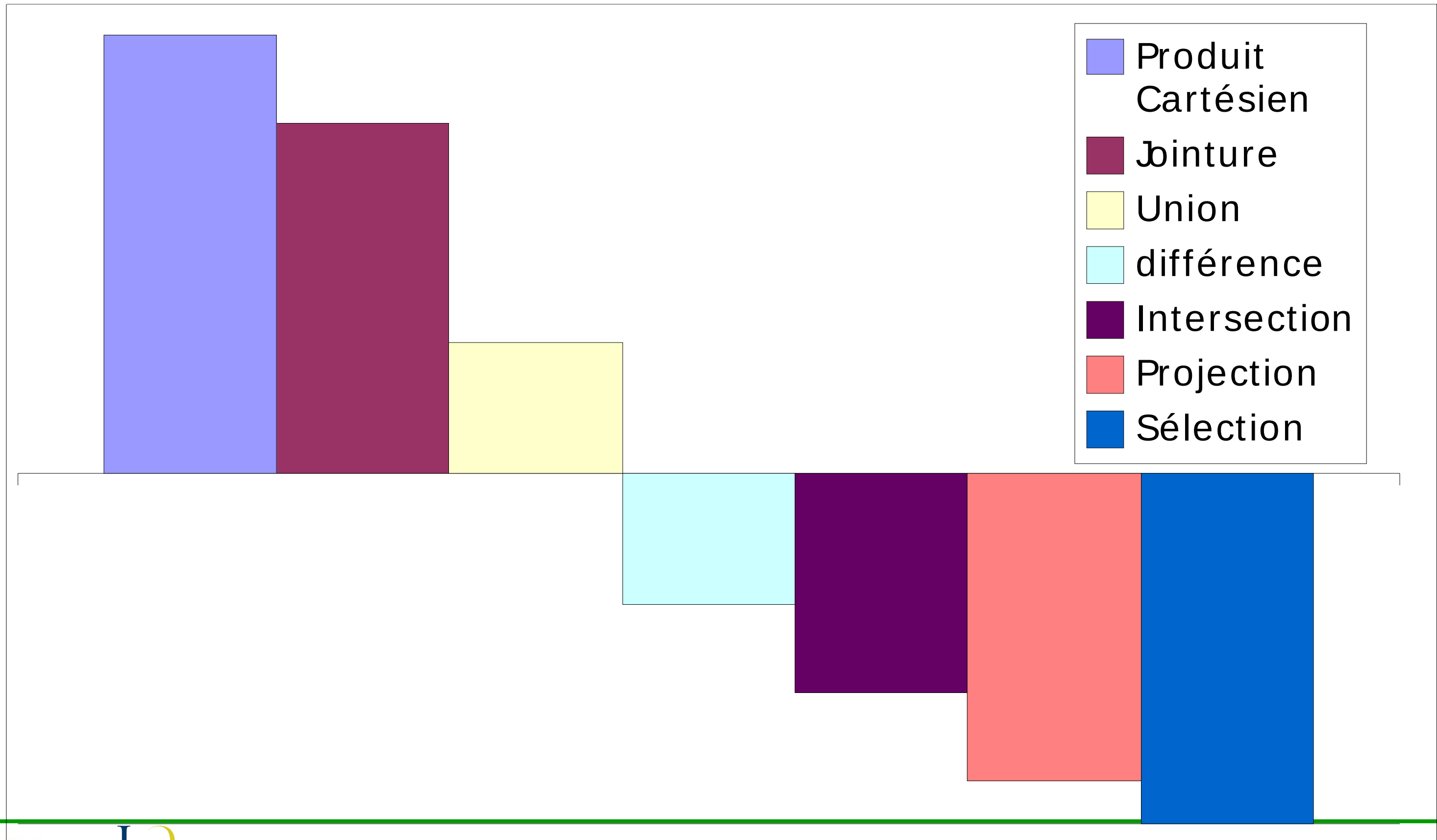
- θ-jointure, Union et Intersection sont commutatives
- θ-jointure, Union et Intersection sont associatives
- $\sigma_{Q_2}(\sigma_{Q_1}(R)) = \sigma_{Q_1 \wedge Q_2}(R)$
- $\sigma_{Q_1}(R) \cup \sigma_{Q_2}(R) = \sigma_{Q_1 \vee Q_2}(R)$
- $\Pi_{A_1,\ldots,A_p}(\sigma_Q(R)) = \sigma_Q(\Pi_{A_1,\ldots,A_p}(R))$
  $\iff \forall A_i \in Q, A_i \in \{A_1, \ldots, A_p\}$
- Si $\exists A_j \in Q / A_j \notin \{A_1, \ldots, A_p\}$, alors
  $\Pi_{A_1,\ldots,A_p}(\sigma_Q(R)) = \Pi_{A_1,\ldots,A_p}(\sigma_Q(\Pi_{A_1,\ldots,A_p,A_j}(R)))$

Propriétés

- $\sigma_{Q(A_i) \wedge Q'(B_j)}(R \bowtie_{Q''} S) = \sigma_{Q(A_i)}(R) \bowtie_{Q''} \sigma_{Q'(B_j)}(S)$
- $\sigma_Q(R \cup S) = \sigma_Q(R) \cup \sigma_Q(S)$
- $\Pi_V(R \bowtie_Q S) =$
  $$\begin{cases} \Pi_{V(R)}(R) \bowtie_Q \Pi_{V(S)}(S) \text{ si } \forall X_j \in Q, X_j \in \{A_i\} \cup \{B_i\} \\ \Pi_V(\Pi_{V(R)\cup\{A_i\}}(R) \bowtie_Q \Pi_{V(S)\cup\{B_i\}}(S)) \text{ sinon} \end{cases}$$
  $\{A_i\} \cup \{B_j\} \subseteq Q$
- $\Pi_V(R \cup S) = \Pi_{V(R)}(R) \cup \Pi_{V(S)}(S)$

Remarque

- Coûts relatifs des opérateurs en terme de quantité d'informations produites (cas général)
  - Taille des tables produites



Remarque

- Coût des opérateurs parfois importants.
- Possibilité de permuter l'ordre des opérations (en fonction des propriétés).
  - Possibilité (nécessité) de réduire considérablement la taille des relations intermédiaires
    - moins de place
    - plus rapide
  - Il existe plusieurs solutions pour obtenir une relation : choisir la plus efficace.
- Piste : en général, cherche à appliquer les opérateurs de sélection et de projection le plus tôt possible.

Un petit exemple

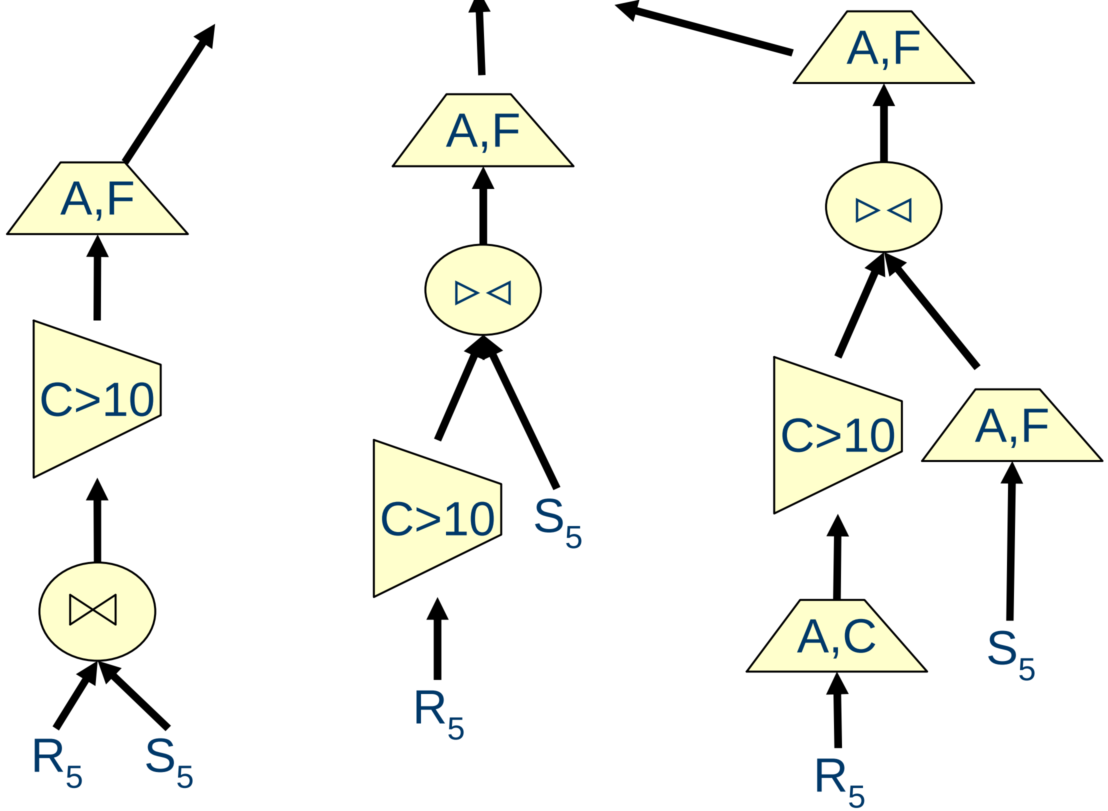
A	F
b	2
b	18
a	100

R5

A	B	C
a	b	2
b	b	12
a	c	20
c	c	1
b	c	18

S5

A	D	E	F
a	k	f	100
b	b	b	2
b	a	b	18



(1) $\prod_{A,F}(\sigma_{c>10}(R_5 \bowtie S_5))$

(2) $\prod_{A,F}(\sigma_{c>10}(R_5) \bowtie S_5)$

(3) $\prod_{A,F}(\sigma_{c>10}(\prod_{A,C}(R_5)) \bowtie \prod_{A,F}(S_5))$

(1)			Taille
30		24	54
	6*7*2		84
	5*7*2		70
	3*2*2		12
	Max		84

	(2)		Taille
30		24	54
3*3*2		24	42
	5*7*2		70
	12		12
	Max		70

	(3)		Taille
30		24	54
5*2*2		2*2*3	32
3*2*2		2*2*3	24
	5*3*2		30
	12		12
	Max		32(54)

Gain 61,90% 35,71%

Résumé des opérateurs

- **Sélection** : relation consistant en l'ensemble de tous les n-uplets d'une certaine relation satisfaisant une certaine condition
- **Projection** : relation consistant en l'ensemble de tous les n-uplets, qui restent présents dans une relation après que certains attributs ont été éliminés
- **Produit** : relation consistant en l'ensemble de tous les n-uplets résultat de la combinaison de deux n-uplets, provenant de deux relations données
- **Union** : relation consistant en l'ensemble de tous les n-uplets apparaissant dans au moins l'une de deux relations données
- **Intersection** : relation consistant en l'ensemble de tous les n-uplets apparaissant dans les deux relations données
- **Différence** : relation consistant en l'ensemble de tous les n-uplets apparaissant dans la première relation donnée, mais pas dans la seconde
- **Jointure** : relation consistant en l'ensemble de tous les n-uplets qui soit une combinaison de deux n-uplets