

**Cours Bases de données**  
**3<sup>ème</sup> Année Systèmes d'Information**

# **Chapitre 03**

## **Le Langage Algébrique**

Fouad DAHAK

Enseignant-Chercheur

Chargé de cours Bases de données

Ecole Nationale Supérieure d'Informatique (ESI)

(f\_dahak@esi.dz – <http://dahak.esi.dz>)

**Table des matières**

I. Algèbre relationnelle .....	3
I.1. Présentation .....	3
I.2. Les opérations ensemblistes .....	3
I.2.1. Union .....	3
I.2.2. Différence .....	3
I.2.3. Produit cartésien .....	4
I.3. Les opérations spécifiques .....	5
I.3.1. Projection .....	5
I.3.2. Restriction (Sélection) .....	6
I.3.3. Thêta Jointure .....	6
I.3.4. Jointure Naturelle .....	7
I.4. Les opérations dérivées .....	8
I.4.1. Intersection .....	8
I.4.3. Jointure externe .....	10
I.4.4. Semi-jointure .....	11
<b>I.4.4.2. Notation</b> .....	11
<b>I.4.4.3. Représentation graphique</b> .....	11
I.5. Opération de renommage .....	11
I.6. L'Affectation .....	12
I.7. La Valeur NULL .....	12
I.8. Modification de la base de données .....	12
I.8.1. Suppression .....	12
I.8.2. Insertion .....	13
I.8.3. Modification .....	13
II. Le langage algébrique .....	13
II.1. Présentation .....	13
II.2. Comment construire une requête algébrique? .....	13
II.3. Arbre algébrique .....	14
II.4. Fonctions et agrégats .....	14
II.4.1. Fonction de calcul .....	14
II.4.2. Les agrégats .....	14

## I. Algèbre relationnelle

### I.1. Présentation

L'algèbre relationnelle a été inventée par E. Codd en 1970 dont le but de formaliser les opérations sur les ensembles. Elle constitue une collection d'opérations formelles qui agissent sur des relations et produisent des relations.

Ces opérations sont regroupées, selon leurs caractéristiques, en plusieurs familles.

### I.2. Les opérations ensemblistes

#### I.2.1. Union

##### I.2.1.1. Définition

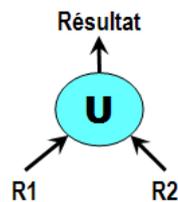
L'union est une opération sur deux relations de même schéma R1 et R2 qui sert à construire une troisième relation R3 de même schéma ayant comme tuples ceux appartenant à R1, à R2 ou aux deux.

Les tuples qui apparaissent plusieurs fois dans le résultat ne sont représentés qu'une seule fois (pas de doublons)

##### I.2.1.2. Notations

$R1 \cup R2$   
**UNION**(R1,R2)  
 APPEND(R1,R2)

##### I.2.1.3. Représentation graphique



#### Exemple

**R<sub>1</sub>**

N°	Nom	TEL
1	ALI	021324354

**U**

**R<sub>2</sub>**

N°	Nom	TEL
2	AHMED	021554354

**=**

**R<sub>3</sub>**

N°	Nom	TEL
1	ALI	021324354
2	AHMED	021554354

### I.2.2. Différence

#### I.2.2.1. Définition

La différence est une opération sur deux relations de même schéma R1 et R2 qui sert à construire une troisième relation R3 de même schéma ayant comme tuples ceux appartenant à R1 et n'appartenant pas à R2.

**I.2.2.2. Notations**

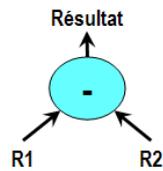
R1 - R2

**DIFFERENCE(R1,R2)**

REMOVE(R1,R2)

MINUS(R1,R2)

**I.2.2.3. Représentation graphique**



**Exemple**

<b>R<sub>1</sub></b>	<b>N°</b>	<b>Nom</b>	<b>TEL</b>
	1	ALI	021324354
<b>-</b>	2	AHMED	021554354
<b>R<sub>2</sub></b>	<b>N°</b>	<b>Nom</b>	<b>TEL</b>
<b>=</b>	2	AHMED	021554354
<b>R<sub>3</sub></b>	<b>N°</b>	<b>Nom</b>	<b>TEL</b>
	1	ALI	021324354

**I.2.3. Produit cartésien**

**I.2.3.1. Définition**

Le produit cartésien de deux relations R1 et R2 de schéma quelconque est une relation R3 ayant pour attributs la concaténation des attributs de R1 et de R2 et dont les tuples sont constitués de toutes les concaténations d'un tuple de R1 à un tuple de R2

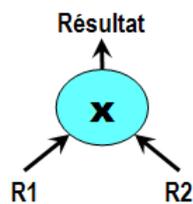
**I.2.3.2. Notations**

R1 x R2

**PRODUCT(R1,R2)**

TIMES(R1,R2)

**I.2.3.3. Représentation graphique**



**Exemple**

<b>R<sub>1</sub></b>	<table border="1"><thead><tr><th>N°</th><th>Nom</th><th>TEL</th></tr></thead><tbody><tr><td>1</td><td>ALI</td><td>021324354</td></tr><tr><td>2</td><td>AHMED</td><td>021554354</td></tr></tbody></table>	N°	Nom	TEL	1	ALI	021324354	2	AHMED	021554354
N°	Nom	TEL								
1	ALI	021324354								
2	AHMED	021554354								

x

<b>R<sub>2</sub></b>	<table border="1"><thead><tr><th>Cmd</th><th>Date</th></tr></thead><tbody><tr><td>2</td><td>23/02/2007</td></tr></tbody></table>	Cmd	Date	2	23/02/2007
Cmd	Date				
2	23/02/2007				

=

<b>R<sub>3</sub></b>	<table border="1"><thead><tr><th>N°</th><th>Nom</th><th>TEL</th><th>CMD</th><th>DATE</th></tr></thead><tbody><tr><td>1</td><td>ALI</td><td>021324354</td><td>2</td><td>23/02/2007</td></tr><tr><td>2</td><td>AHMED</td><td>021554354</td><td>2</td><td>23/02/2007</td></tr></tbody></table>	N°	Nom	TEL	CMD	DATE	1	ALI	021324354	2	23/02/2007	2	AHMED	021554354	2	23/02/2007
N°	Nom	TEL	CMD	DATE												
1	ALI	021324354	2	23/02/2007												
2	AHMED	021554354	2	23/02/2007												

**I.2.3.5. Note**

Dans le cas où les deux opérations ont des attributs ayant les même noms, on représente au niveau du résultat ces attributs avec d'autres noms ou bien en spécifiant la relation à laquelle ils appartiennent : R1.A, R2.A.

**I.3. Les opérations spécifiques**

**I.3.1. Projection**

**I.3.1.1. Définition**

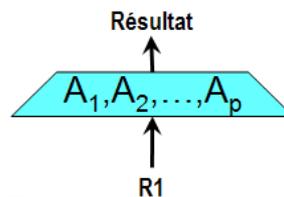
La projection d'une relation R(A1,A2,...,An) sur les attributs Ai,Ai+1,...,Ap (avec p<n) est une relation R2 de schéma Ai,Ai+1,...,Ap et dont les tuples sont obtenus par élimination des attributs de R n'appartenant pas à R2 et par suppression des doublons.

**I.3.1.2. Notations**

$P_{A_1,A_2,\dots,A_p}(R)$   
 $R[A_1,A_2,\dots,A_p]$

**PROJECT(R,A1,A2,...,Ap)**

**I.3.1.3. Représentation graphique**



**Exemple**

<b>R</b>	<table border="1"><thead><tr><th>N°</th><th>Nom</th><th>TEL</th></tr></thead><tbody><tr><td>1</td><td>ALI</td><td>021324354</td></tr><tr><td>2</td><td>AHMED</td><td>021554354</td></tr></tbody></table>	N°	Nom	TEL	1	ALI	021324354	2	AHMED	021554354
N°	Nom	TEL								
1	ALI	021324354								
2	AHMED	021554354								

$\Pi_{Nom}(R)$	<table border="1"><thead><tr><th>Nom</th></tr></thead><tbody><tr><td>ALI</td></tr><tr><td>AHMED</td></tr></tbody></table>	Nom	ALI	AHMED
Nom				
ALI				
AHMED				

### I.3.2. Restriction (Sélection)

#### I.3.2.1. Définition

La restriction (ou sélection) de la relation R par une condition C est une relation R2 de même schéma dont les tuples sont ceux de R satisfaisant la condition C.

La condition est de la forme <Attribut>Opérateur<Valeur>

Les opérateurs sont {=,<,>,<=,>=,<>}

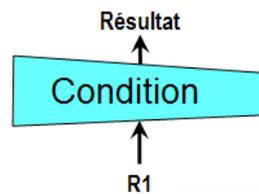
#### I.3.2.2. Notations

$S_{Condition}(R)$

$R[Condition]$

**RESTRICT(R,Condition)**

#### I.3.2.3. Représentation graphique



#### Exemple

N°	Nom	TEL
1	ALI	021324354
2	AHMED	021554354
3	ALI	034456543

$$\sigma_{nom = 'ALI'}(R)$$

N°	Nom	TEL
1	ALI	021324354
3	ALI	034456543

### I.3.3. Thêta Jointure

#### I.3.3.1. Définition

La thêta-jointure de deux relations R1 et R2 de schéma quelconque selon une condition C est une relation R3 dont le schéma est la concaténation des attributs des deux relations et les tuples sont ceux du produit cartésien entre R1 et R2 respectant la condition C.

La condition C est de la forme <Attribut>opérateur<Attribut>

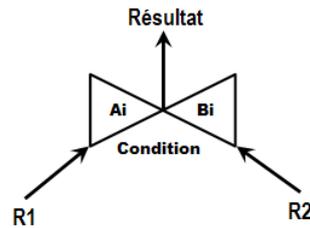
Les opérateurs peuvent être arithmétiques (=,>,<,>=,<=,<>) ou logique (Et, Ou,Non)

#### I.3.3.2. Notations



**JOIN(R1,R2,Condition)**

#### I.3.3.3. Représentation graphique



Exemple

**R1**

N	Nom	TEL
1	ALI	021324354
2	AHMED	021554354
3	ALI	034456543

**R2**

Num	Nom	Adresse
0	ALI	Alger
2	AHMED	Blida
5	MALIK	Oran



N	Nom	TEL	Num	Nom	Adresse
1	ALI	021324354	0	ALI	Alger
2	AHMED	021554354	0	ALI	Alger
2	AHMED	021554354	2	AHMED	Blida
3	ALI	034456543	0	ALI	Alger
3	ALI	034456543	2	AHMED	Blida

### 1.3.3.5. Note

Si l'opérateur est « = » Alors c'est une **Equi-jointure**  
Sinon c'est une **Inéqui-jointure**

### 1.3.4. Jointure Naturelle

#### 1.3.4.1. Définition

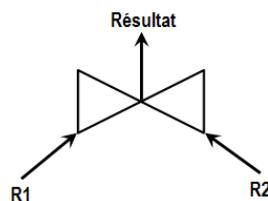
La jointure naturelle de deux relations R1 et R2 de schéma quelconque donne une troisième relation R3 dont le schéma est obtenu avec concaténation des attributs de R1 et ceux de R2 mais en ne prenant les attributs de même nom qu'une seule fois. Les tuples de R3 sont ceux de R1 et de R2 respectant une equi-jointure entre les attributs de même nom.

#### 1.3.4.2. Notation

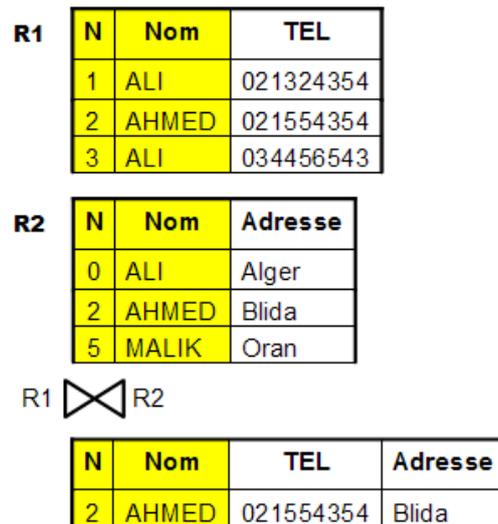


$JOIN(R1,R2)$

#### 1.3.4.3. Représentation graphique



Exemple



**1.3.4.5. Note**

Une jointure naturelle entre deux relations R1 et R2 n'ayant aucun attribut en commun (de même nom) est le produit cartésien de R1 et de R2.

**I.4. Les opérations dérivées**

**I.4.1. Intersection**

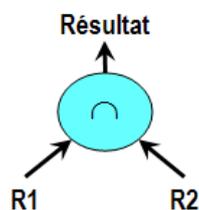
**I.4.1.1. Définition**

L'intersection de deux relation R1 et R2 de même schéma est une relation R3 de même schéma dont les tuples sont ceux appartenant à la fois à R1 et à R2.

**I.4.1.2. Notation**

R1  $\cap$  R2  
**INTERSECT(R1,R2)**  
 AND(R1,R2)

**I.4.1.3. Représentation graphique**



Exemple

$R_1$	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">N°</th> <th style="width: 40%;">Nom</th> <th style="width: 50%;">TEL</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>ALI</td> <td>021324354</td> </tr> <tr style="background-color: yellow;"> <td>2</td> <td>AHMED</td> <td>021554354</td> </tr> </tbody> </table>	N°	Nom	TEL	1	ALI	021324354	2	AHMED	021554354
N°	Nom	TEL								
1	ALI	021324354								
2	AHMED	021554354								
$\cap$										
$R_2$	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">N°</th> <th style="width: 40%;">Nom</th> <th style="width: 50%;">TEL</th> </tr> </thead> <tbody> <tr style="background-color: yellow;"> <td>2</td> <td>AHMED</td> <td>021554354</td> </tr> </tbody> </table>	N°	Nom	TEL	2	AHMED	021554354			
N°	Nom	TEL								
2	AHMED	021554354								
$=$										
$R_3$	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">N°</th> <th style="width: 40%;">Nom</th> <th style="width: 50%;">TEL</th> </tr> </thead> <tbody> <tr style="background-color: yellow;"> <td>2</td> <td>AHMED</td> <td>021554354</td> </tr> </tbody> </table>	N°	Nom	TEL	2	AHMED	021554354			
N°	Nom	TEL								
2	AHMED	021554354								

### 1.4.2. Division (Quotient)

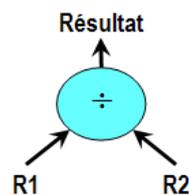
#### 1.4.2.1. Définition

La division de la relation  $R(A_1, A_2, \dots, A_n)$  par la sous-relation  $R_2(A_{p+1}, \dots, A_n)$  est la relation  $R_3(A_1, A_2, \dots, A_p)$  formées de tous les tuples qui concaténés à chaque tuple de  $R_2$  donnent toujours un tuples de  $R_1$ .

#### 1.4.2.2. Notation

$R_1 / R_2$   
***DIVISION(R1,R2)***

#### 1.4.2.3. Représentation graphique



Exemple

$R_1$	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">Nom</th> <th style="width: 15%;">Age</th> <th style="width: 75%;">Ville</th> </tr> </thead> <tbody> <tr> <td>ALI</td> <td>12</td> <td>ALGER</td> </tr> <tr> <td>ALI</td> <td>23</td> <td>ORAN</td> </tr> <tr style="background-color: yellow;"> <td>AHMED</td> <td>12</td> <td>ALGER</td> </tr> <tr style="background-color: yellow;"> <td>AHMED</td> <td>23</td> <td>ALGER</td> </tr> <tr> <td>MALIK</td> <td>43</td> <td>ALGER</td> </tr> </tbody> </table>	Nom	Age	Ville	ALI	12	ALGER	ALI	23	ORAN	AHMED	12	ALGER	AHMED	23	ALGER	MALIK	43	ALGER
Nom	Age	Ville																	
ALI	12	ALGER																	
ALI	23	ORAN																	
AHMED	12	ALGER																	
AHMED	23	ALGER																	
MALIK	43	ALGER																	
$\div$																			
$R_2$	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 10%;">Age</th> <th style="width: 90%;">Ville</th> </tr> </thead> <tbody> <tr> <td>12</td> <td>ALGER</td> </tr> <tr> <td>23</td> <td>ALGER</td> </tr> </tbody> </table>	Age	Ville	12	ALGER	23	ALGER												
Age	Ville																		
12	ALGER																		
23	ALGER																		
$=$																			
$R_3$	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 100%;">Nom</th> </tr> </thead> <tbody> <tr> <td>AHMED</td> </tr> </tbody> </table>	Nom	AHMED																
Nom																			
AHMED																			

**1.4.2.5. Note**

Les attributs du résultat d'une division sont ceux faisant partie de la première relation et ne sont pas dans la seconde pour que le produit cartésien du résultat avec la deuxième donnent tous les attributs de la première relation.

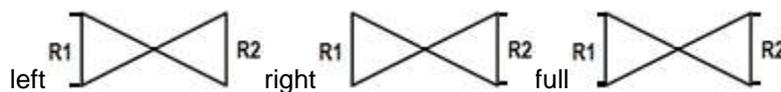
Pour effectuer une division entre R1 et R2 il faut que tous les attributs de R2 font partie de R1 et que R1 possède au moins un attribut en plus que R2.

**1.4.3. Jointure externe**

**1.4.3.1. Définition**

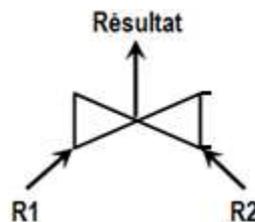
La jointure externe entre deux relations R1 et R2 de schéma quelconque est une relation R3 dont le schéma est la concaténation des attributs de R1 et de ceux de R2 en ne représentant les attributs ayant le même nom qu'une seule fois. Les tuples de R3 sont ceux obtenus avec une jointure naturelle entre R1 et R2 et ceux de R1 et de R2 ne participants pas à la jointure en représentant par des valeurs nulles ceux de l'autre relation.

**1.4.3.2. Notation**



**EXT-JOIN(R1,R2)**

**1.4.3.3. Représentation graphique**



Exemple

**R1**

NA	Nom	TEL
1	ALI	021324354
2	AHMED	021554354
3	ALI	034456543

**R2**

NB	Nom	Adresse
0	ALI	Alger
2	MALIK	Blida
5	MALIK	Oran



NA	Nom	TEL	NB	Adresse
1	ALI	021324354	0	Alger
2	AHMED	021554354	-	-
3	ALI	034456543	0	Alger
-	MALIK	-	2	Blida
-	MALIK	-	5	Oran

### I.4.3.5. Note

On distingue deux autres variantes de la jointure externe, la jointure externe droite et la jointure externe gauche notées respectivement REXT-JOIN et LEXT-JOIN. La première donne tous les attributs de la relation à droite de la jointure externe et uniquement ceux de la relation gauche qui participent à la jointure. La seconde c'est l'inverse.

### I.4.4. Semi-jointure

#### I.4.4.1. Définition

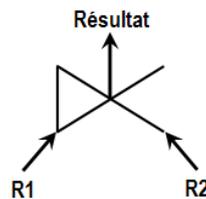
La semi-jointure deux relations R1 et R2 de schéma quelconque est une relation R3 dont le schéma est celui de R1 et les tuples sont ceux de R1 appartenant à la jointure naturelle entre R1 et R2.

#### I.4.4.2. Notation

$$R1 \bowtie R2$$

$$\text{SEMI-JOIN}(R1,R2)$$

#### I.4.4.3. Représentation graphique



Exemple

**R1**

NA	Nom	TEL
1	ALI	021324354
2	AHMED	021554354
3	ALI	034456543

**R2**

NB	Nom	Adresse
0	ALI	Alger
2	MALIK	Blida
5	MALIK	Oran

$$R1 \bowtie R2$$

NA	Nom	TEL
1	ALI	021324354
3	ALI	034456543

### I.5. Opération de renommage

Pourquoi renommer ?

1. Le résultat d'une expression algébrique ne possède pas de nom;

2. On a besoin de renommer certains attributs d'une relation ou d'une expression algébrique.

$\rho_{NA \leftarrow \text{num}}(\text{Acteur})$

$\rho_{\text{Acteur\_Ali}}(\sigma_{\text{nom=ali}}(\text{Acteur}))$

### I.6. L'Affectation

Le même principe qu'une affectation dans l'algorithmique.

C'est le fait d'attribuer le résultat d'une expression algébrique à une variable temporaire qu'est dans ce cas une relation intermédiaire.

$R1 \leftarrow \sigma_{\text{taille}=32}(\text{VESTE})$

$R2 \leftarrow \sigma_{\text{couleur=rouge}}(\text{VESTE})$

$R3 \leftarrow R1 \cap R2$

$\text{Result} = \pi_{\text{marque}}(R3)$

### I.7. La Valeur NULL

NULL = « Value unknown or nonexistent »

1. Toute opération arithmétique (+, -, \*, /) avec une valeur NULL renvoie NULL.
2. Toute comparaison (>, <, =, >=, <=, <>) avec la valeur NULL renvoie une valeur UNKNOWN.
3. Les opérateurs booléens se comportent comme suit :

a. **and** : (true and unknown) = unknown  
 (false and unknown) = false  
 (unknown and unknown) = unknown

b. **or** : (true or unknown) = true  
 (false or unknown) = unknown  
 (unknown or unknown) = unknown

c. **not** : (not unknown) = unknown

Quel est le résultat des opérations algébriques appliquées à une valeur NULL?

1. **Restriction** : Si le résultat de la condition est true la ligne est affichée sinon (false ou unknown) la ligne n'est pas retournée dans le résultat.
2. **Jointure** : La jointure est un produit cartésien suivi d'une restriction. Le résultat est le même que pour la restriction.
3. **Projection** : La projection traite le NULL comme une valeur quelconque.
4. **Union, Intersection, différence** : Idem que la projection.
5. **Agrégation** : Idem que la projection.

### I.8. Modification de la base de données

#### I.8.1. Suppression

$r \leftarrow r - E$  avec E une expression algébrique dont le résultat doit être de même schéma que la relation r. le résultat de l'opération est le même que celui de la différence entre r et E.

### I.8.2. Insertion

$r \leftarrow r \cup E$  avec  $E$  une expression algébrique dont le résultat est de même schéma que  $r$ . le résultat est le même que celui de l'union.

### I.8.3. Modification

$r \leftarrow \pi_{f_1, f_2, \dots, f_n}(E_r)$  avec  $E_r$  une expression algébrique sur  $r$ . uniquement les colonnes spécifiées dans la projection qui sont modifiées.

## II. Le langage algébrique

### II.1. Présentation

C'est un langage d'interrogation de bases de données qui est à la base du langage SQL. Les opérations de base de l'algèbre relationnelle constituent un langage complet.

#### Exemple

ACTEUR(**NA**,NOM,PRENOM,ADRESSE,SEXE)  
VESTE(**NV**,MARQUE,COULEUR,TAILLE)  
PORTE(**NA,NV,DATE**,DUREE)

**Q1** : Donner les marques des vestes de taille 32 et de couleur rouge

R1 = RESTRICT(VESTE,TAILLE=32)  
R2 = RESTRICT(VESTE,COULEUR='ROUGE')  
R3 = INTERSECT(R1,R2)  
RESULT=PROJECT(R3,MARQUE)

**Q2** : Donner les noms et prénoms des acteurs qui ont mis des vestes rouges ou bleues

R1 = RESTRICT(VESTE,COULEUR='ROUGE')  
R2 = RESTRICT(VESTE,COULEUR='BLEU')  
R3 = UNION(R1,R2)  
R4 = JOIN(R3,PORTE)  
R5 = JOIN(R4,ACTEUR)  
RESULT=PROJECT(R5,NOM,PRENOM)

**Q3** : Donner les noms et prénoms des acteurs qui ont mis des vestes de taille 32 plus de deux heures, avec la marque de la veste.

R1 = RESTRICT(VESTE,TAILLE=32)  
R2 = RESTRICT(PORTE,DUREE>2)  
R3 = JOIN(R1,R2)  
R4 = PROJECT(R3,NA,MARQUE)  
R5 = JOIN(R4,ACTEUR)  
RESULT=PROJECT(R5,NOM,PRENOM,MARQUE)

### II.2. Comment construire une requête algébrique?

1. Identifier les relations utiles pour exprimer la requête,
2. Recopier le schéma de ces relations, et indiquer sur ces schémas:
  1. les attributs qui font partie du résultat de la requête
  2. les conditions portant sur les attributs
  3. les liens entre les relations
3. Traduire cette figure en expression algébrique
  1. faire les sélections selon les conditions portant sur les attributs,
  2. faire les jointures (naturelles ou thêta) selon les liens entre les relation (une jointure par lien)
  3. projeter sur les attributs qui font partie du résultat

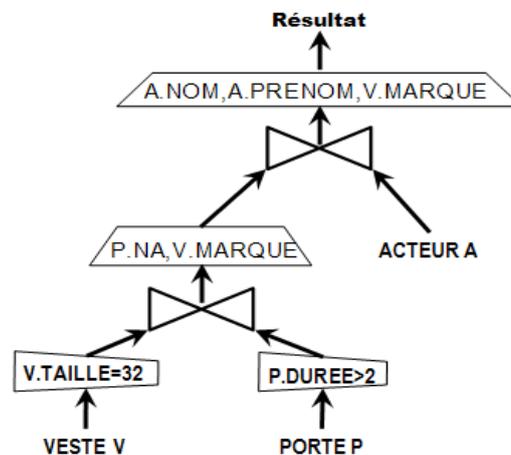
Note

Cette méthode est valable pour la plupart des requêtes. Cependant, certains types de requêtes nécessitent de compliquer la méthode. C'est le cas des requêtes où la même relation est utilisée plusieurs fois avec des ensembles de tuples différents.

### II.3. Arbre algébrique

Arbre dont les nœuds représentent les opérations algébriques et les arcs les relations de base ou temporaires représentant des flots de données entre opérations.

Exemple



### II.4. Fonctions et agrégats

#### II.4.1. Fonction de calcul

On peut remplacer les attributs par des expressions d'attributs.

Une fonction de calcul est une expression arithmétique construite à partir d'attributs d'une relation et de constantes, par application de fonctions arithmétique successives.

Exemple

$R1 = JOIN(VESTE, PORTE, TAILLE * DUREE > DUREE / 3)$

$R2 = RESTRICT(R1, DUREE * 100 / TAILLE > 38)$

$RESULT = PROJECT(R2, NOM, TAILLE - TAILLE * DUREE / 100)$

#### II.4.2. Les agrégats

Les agrégats sont utilisés pour effectuer des opérations de calcul sur des colonnes.

Les fonctions les plus proposées sont : Somme (SUM), Moyenne (AVG), Minimum (MIN), Maximum (Max), Compte (Count)

**Exemple**  $Résultat = Agregat(Veste, Marque, AVG(Taille))$

$Résultat = AVG_{Marque}(Veste, Taille)$

Représentation graphique

