Chapitre : Algèbre Relationnelle

Introduction :

La représentation d'information sous forme relationnelle est intéressante car les fondements mathématiques du relationnel, outre qu'ils permettent une modélisation logique simple et puissante, fournissent également un ensemble de concepts pour manipuler formellement l'information ainsi modélisée.

L'algèbre relationnelle est un ensemble d'opérateurs qui, à partir d'une ou deux relations existantes, créent en résultat une nouvelle relation temporaire. La relation résultat a exactement les mêmes caractéristiques qu'une relation de la base de données et peut donc être manipulée de nouveau par les opérateurs de l'algèbre.

De nombreux opérateurs relationnels ont été proposés, on peut cependant présenter ici les plus courants. On peut classifier les opérateurs relationnels en trois catégories :

- les opérateurs unaires : renommage, sélection et projection

- les opérateurs binaires travaillant sur des relations de même schéma : union, intersection, différence

- les opérateurs binaires travaillant sur des relations de schémas différents : jointure, produit cartésien, théta-jointure, division

Les opérations de l'Algèbre Relationnelle sont notées algébriquement ou représentées graphiquement. Les notations ne sont pas standardisées en algèbre relationnelle. Les notations courantes (pas forcément universelles) sont les suivantes.

Les opérateurs de l'algèbre peuvent être regroupés en deux classes :

- les opérateurs provenant de la théorie mathématique sur les ensembles (applicables car chaque relation est définie comme un ensemble de tuples): *union*, *intersection*, *différence*, *produit*;
- les opérateurs définis spécialement pour les bases de données relationnelles: *sélection*, *projection*, *jointure*, *division* et *renommage*.

Les opérations de l'Algèbre Relationnelle sont notées algébriquement ou représentées graphiquement. Les notations ne sont pas standardisées en algèbre relationnelle. Les notations courantes (pas forcément universelles) sont les suivantes.

1- **Sélection : Notation σ** (σ est la lettre grecque sigma.)

❖ Opération unaire dont la syntaxe est la suivante : σp(R)où p est un prédicat (expression logique ou condition de sélection)

- ❖ Rôle : La sélection génère une nouvelle relation de même schéma que la relation R, regroupant exclusivement tous les tuples de R qui satisfont l'expression logique p.
- ❖ En d'autres termes, la sélection permet de choisir des lignes dans la table R.
- ❖ Si R est vide (c'est-à-dire ne contient aucune occurrence(tuple)), la relation qui résulte de la sélection est vide.

Exemple :

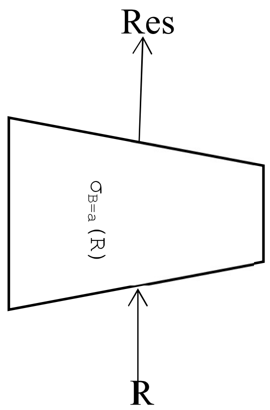
R	A	B
	1	a
	2	b
	3	a

- Res := $\sigma_{B=a}$ (R)

Res	A	B
	1	a
	3	a

$\sigma_{B=a}$ (R)

condition logique



Représentation graphique

2- Projection : Notation π (π est la lettre grecque pi.)

- o Opération unaire essentielle dont la syntaxe est la suivante : $\pi_{\mathbf{A}}R$ (A est un sous ensemble des attributs de R).
- o **Rôle :** La projection génère une relation regroupant exclusivement toutes les occurrences de la relation **R** réduites aux attributs de la liste d'attributs **A**.
- o En d'autres termes, la projection permet de choisir des colonnes dans la table R.
- o Le résultat de la projection est une nouvelle relation dont les attributs sont ceux présents dans **A**.
- o Si R est vide, la relation qui résulte de la projection est vide.

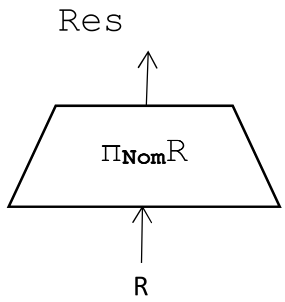
Remarque : La projection est une Opération sur une RELATION 1 consistant à composer une relation RELATION2 en enlevant à la relation initiale tous les attributs non mentionnés en opérandes (aussi bien au niveau du schéma que des tuples) et en éliminant les tuples en double qui sont conservés une seule fois.

Exemple :

R	Numero	Nom
	1	a
	2	b
	3	a

$\pi_{\textbf{Nom}}R$

Res	Nom
	a
	b



Représentation graphique

3- Union: Notation **R1 U R2**

- L'union est une opération binaire ensembliste qui génère une relation regroupant toutes les occurrences de la relation R1 et toutes les occurrences de la relation R2.

- Si une même occurrence existe dans R1 et dans R2, elle n'apparaît qu'une seule fois dans le résultat de l'union.
- **Contrainte :** R1 et R2 doivent avoir le même schéma (les mêmes attributs).
- Le résultat de l'union est une nouvelle relation qui a les mêmes attributs que R1 et R2.
- Si R1 et R2 sont vides, la relation qui résulte de l'union est vide. Si R1 (respectivement R2) est vide, la relation qui résulte de l'union est identique à R2 (respectivement R1).

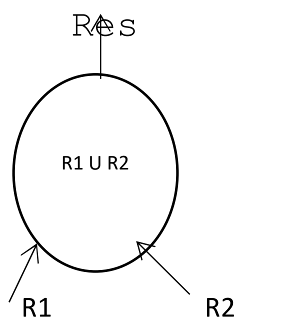
Exemple :

R1	Numero	Nom
	1	a
	2	b
	3	a

R2	Numero	Nom
	1	a
	2	c
	4	d

Res	Numero	Nom
	1	a
	2	b
	3	a
	2	c
	4	d

Res := R1 U R2



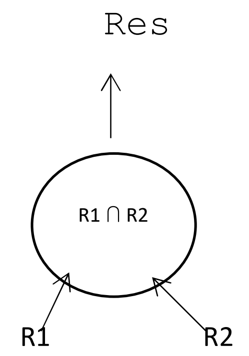
Représentation Graphique

4- **Intersection:** Notation **R1 ∩ R2**

- L'intersection est une opération binaire ensembliste qui génère une relation regroupant exclusivement toutes les occurrences qui existent à la fois dans la relation R1 et dans la relation R2.
- **Contrainte :** R1 et R2 doivent avoir le même schéma (les mêmes attributs).
- Le résultat de l'intersection est une nouvelle relation qui a les mêmes attributs que R1 et R2.
- Si R1 et R2 sont vides, la relation qui résulte de l'intersaction est vide.

`Res := R1∩R2`

Res	Numero	Nom
	1	a



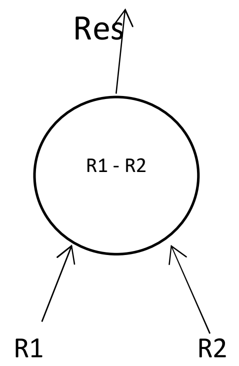
Représentation Graphique

5- **Difference:** Notation **R1 - R2**

- La différence est une opération binaire ensembliste non commutative qui génère une relation regroupant exclusivement toutes les occurrences de la relation R1 qui n'existent pas dans la relation R2.
- **Contrainte :** R1 et R2 doivent avoir les mêmes attributs.
- Le résultat de la différence est une nouvelle relation qui a les mêmes attributs que R1 et R2.
- Si R1 est vide, la relation qui résulte de la différence est vide. Si R2 est vide, la relation qui résulte de la différence est identique à R1.

Res := R1 - R2

Res	Numero	Nom
	2	b
	3	a



Représentation Graphique

6- Produit cartésien : Notation R1x R2

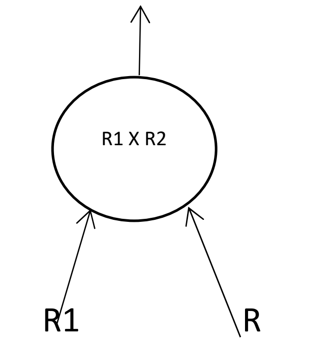
- Le produit cartésien est une opération binaire commutative génère une relation toutes les possibilités de combinaison des occurrences des relations `R1` et `R2`
  - **Contrainte :** `R1` et `R2` ne doivent avoir aucun attribut commun.
  - Le résultat du produit cartésien est une nouvelle relation qui a tous les attributs de `R1` et tous ceux de `R2`.
    - Le nombre d'occurrences de la relation qui résulte du produit cartésien est le nombre d'occurrences de `R1` multiplié par le nombre d'occurrences de `R2`

R1	A	B
	1	a
	2	b

R2	C
	1
	2

Res := R1 X R2

Res	A	B	c
	1	a	1
	1	a	2
	2	b	1
	2	b	2



Représentation Graphique

7- Division

La ***division*** permet de rechercher dans une relation les sous-tuples qui sont complétés par tous ceux d'une autre relation. Elle permet ainsi d'élaborer la réponse à des questions de la forme "quel que soit x, trouver y" de manière simpl

Définition : Soient deux relations R (A_1, …, A_n) et V (A_1, …, A_f) ($f < n$) telles que tous les attributs de V sont aussi attributs de R, alors la division de R par V, notée :

R / V crée une nouvelle relation temporaire de schéma (A_{f+1}, A_{f+2}, …, A_n), et de population égale aux tuples de R, tronqués à [A_{f+1}, A_{f+2}, …, A_n], et qui existent dans R concaténés à tous les tuples de V, c'est-à-dire :

$\{<a_{f+1}, a_{f+2}, ..., a_n> / \forall <a_1, a_2, ... a_f> \in V, \exists <a_1, a_2, ... a_f, a_{f+1}, a_{f+2}, ..., a_n> \in R$

8- Exemple: Division

R	A	B		C
	1	1		1
	1	2		0
	1	3		0
	2	1		1
	2	3		3
	3	1		1
	3	2		0
	3	2		1
	1	2		1

V	B	C
	1	1
	2	0

R/V	A
	1
	3

V’	B	C
	1	1

R/V’	A
	1
	2
	3

V’’	B	C
	2	4

R/V’’	A
	/

9- Renommer un ou des attributs d'une relation

L'opérateur renommer, noté α, permet de changer le nom d'un (ou plusieurs) attribut d'une relation R:

α [nom_attr1: nouveau_nom_pour_attr1, …] R

Cet opérateur est utile avant les jointures s'il y a un problème d'homonymie ou de synonymie, ou avant les opérations ensemblistes (union, différence, intersection) qui requièrent que les attributs correspondants aient le même nom.

10- JOINTURE

La jointure est une des opérations essentielles de l’algèbre relationnelle, sans doute la plus difficile à réaliser dans les systèmes. La jointure permet de composer deux relations à l’aide d’un critère de jointure. Elle peut être vue comme une extension du produit cartésien avec une condition permettant de comparer des attributs. Nous la définirons comme suit:

Opération consistant à rapprocher selon une condition, les tuples de deux relations RELATION1 et RELATION2 afin de former une troisième relation RELATION3 qui contient l’ensemble de tous les tuples obtenus en concaténant un tuple de RELATION1 et un tuple de RELATION2 vérifiant la condition de rapprochement.

La jointure de deux relations produit donc une troisième relation qui contient toutes les combinaisons de tuples des deux relations initiales qui satisfont la condition spécifiée. La condition doit bien sûr permettre le rapprochement des deux relations, et donc être du type:

<Attribut 1> <opérateur> <Attribut2>

où Attribut1 appartient à RELATION1 et Attribut2 à RELATION2.

Selon le type d'opérateur, on distingue:

— l'équi-jointure dans le cas où l'opérateur est =, qui est une véritable composition de relations au sens mathématique du terme

— l'inéqui-jointure dans les autres cas, c'est-à-dire avec un des opérateurs $<, <, >, > \neq$

Dans le cas d'équi-jointure, les deux attributs égaux apparaissent chacun dans le résultat: il y a donc duplication d'une même valeur dans chaque tuple. Afin d'éliminer cette redondance, on définit la jointure naturelle comme suit:

Opération consistant à rapprocher selon une condition les tuples de deux relations RELATION1 et RELATION2 afin de former une troisième relation RELATION3 dont les attributs sont l'union des attributs RELATION1 et de RELATION2 dont les tuples sont obtenus en composant un tuple de RELATION1 et un TUPLE de RELATION2 ayant même valeurs pour les attributs de même nom.

L’opération de jointure est représentée par l’une des notations suivantes, la condition étant simplement omise dans le cas de jointure naturelle (c’est alors l’égalité des attributs de même nom):

- RELATION 1 $\underset{\text{Condition}}{\bowtie}$ RELATION2
- JOIN (RELATION1, RELATION2, Condition)

Jointure (naturelle) de deux relations ayant au moins un attribut **commun.**

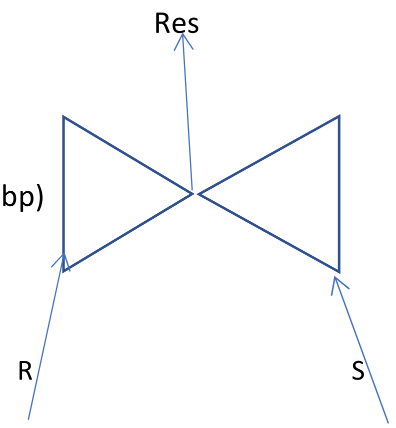
Définition : étant donné deux relations R(X, Y) et S(Y, Z), où X, Y, Z symbolisent soit un attribut, soit un ensemble d'attributs, et où Y n'est pas vide, la jointure (naturelle) de R et S, notée :

$$R \bowtie S$$

crée une nouvelle relation temporaire, de schéma (X, Y, Z). La population de R ▷◁ S est l'ensemble des tuples <x, y, z> créés par composition d'un tuple <x, y> de R et d'un tuple <y, z> de S, tels que les deux tuples ont la même valeur pour Y.

On remarque que la population de R ▷◁ S comporte n tuples, $n \in [0 : card(R) \times card(S)]$, les valeurs extrêmes étant obtenues dans les cas suivants :
- 0 : il n'existe pas de tuple de R et S qui ont même valeur pour Y,
- card(R) x card(S) : les tuples de R et de S ont tous la même valeur, y_0, pour Y.

- But
  - Permet d’établir le lien sémantique entre les relations
- Contraintes
  - Binaire
  - Schéma du résultat:
    - R(a1, a2, …., an), S(b1, b2, ..., bp)
    - Res := R ▷◁ S T(a1, a2, …., an, b1, b2, ..., bp)
- Notation
  - Notation textuelle: T := R ▷◁ S



Notation graphique

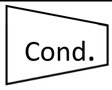
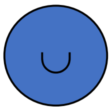
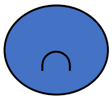
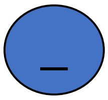
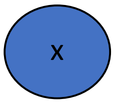
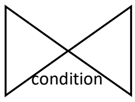
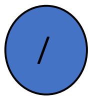
Jointure : Exemple

R	B	C
	1	1
	2	0
	3	8

S	B	D
	1	5
	3	0
	1	9

$R \bowtie S$	B	C	D
	1	1	5
	1	1	9
	3	8	0

Sémantique et notations des opérateurs

Opérateur	**Sémantique**	**Notation textuelle**	**Notation graphique**
Restriction	« Sélectionner » des tuples	$T := \sigma_{cond}(R)$	
Projection	« Sélectionner » des attributs	$T := \Pi_{attributs}(R)$	
Union	Fusionner les extensions de 2 relations	$T := R \cup S$	
Intersection	Obtenir l'ensemble des tuples communs à deux relations	$T := R \cap S$	
Différence	Tuples d'une relation qui ne figurent pas dans une autre	$T := R - S$	
Produit cartésien	Concaténer chaque tuple de R avec chaque tuple de S	$T := R \times S$	
Jointure	Etablir le lien sémantique entre les relations	$T := R \underset{condition}{\rhd\!\lhd} S$	
Division	Répondre aux requêtes de type « tous les »	$T := R / S$	

Exercice

Soit le schéma relationnel suivant :

LIVRE (**Code**, Titre, Auteur, ISBN)

ETUDIANT (**Matricule**, Nom, Date_Nais)

EMPRUNT (***Code***, ***Matricule***, Date_Emprunt, Date_retour)

Expressions d'algèbre

Les opérateurs de l'algèbre peuvent être combinés dans des expressions pour exprimer des requêtes non élémentaires.
Exemple : on obtient la liste des matricule et noms et des étudiants nés avant 2000 par l'expression :Res := π [nom, Matricule] $\sigma_{[date\_nais< '01/01/2000']}$ ETUDIANTS

Le même résultat pourrait être obtenu en écrivant deux opérations l'une après l'autre en créant une relation intermédiaire (dans ce cas il faut nommer les relations intermédiaires) :

Res1 := $\sigma_{[date\_nais<'01/01/2000']}$ ETUDIANTS

Res:= π [nom, Matricule] Res1

Un exemple des populations de chaque relation.

LIVRE	**Code**	Titre	Auteur	ISBN
	1A	Data Bases	Date	13254
	2B	JAVA EE	Thierry Richard	201712
	3C	Les réseaux	Guy Pujolle	45613

ETUDIANTS	**Matricule**	Nom	Date_Nais
	111_IN	Ben Ali	01/02/2000
	342_MI	Amiar	12/12/1999
	231_IN	Salmi	03/06/2001
	324_PH	Ben Saada	15/10/1998

EMPRUNT	**Code**	**Matricule**	Date_Emprunt	Date_Retour
	1A	231_IN	10/10/2019	25/10.2019
	1A	342_MI	10/10/2019	25/10.2019
	3C	111_IN	21/11/2019	04/12/2019
	3C	342_MI	01/12/2019	15/12/2019
	1A	111_IN	04/01/2020	19/01/2020
	2B	111_IN	05/02/2020	20/02/2020

Exprimer les requêtes suivantes en AR :

1- Titres des livres écrits par l'auteur 'Date'
2- Nom des étudiants qui ont emprunté le livre 'Data Bases
3- Livres qui n'ont jamais été empruntés
4- Livres qui sont empruntés par tous les étudiants.

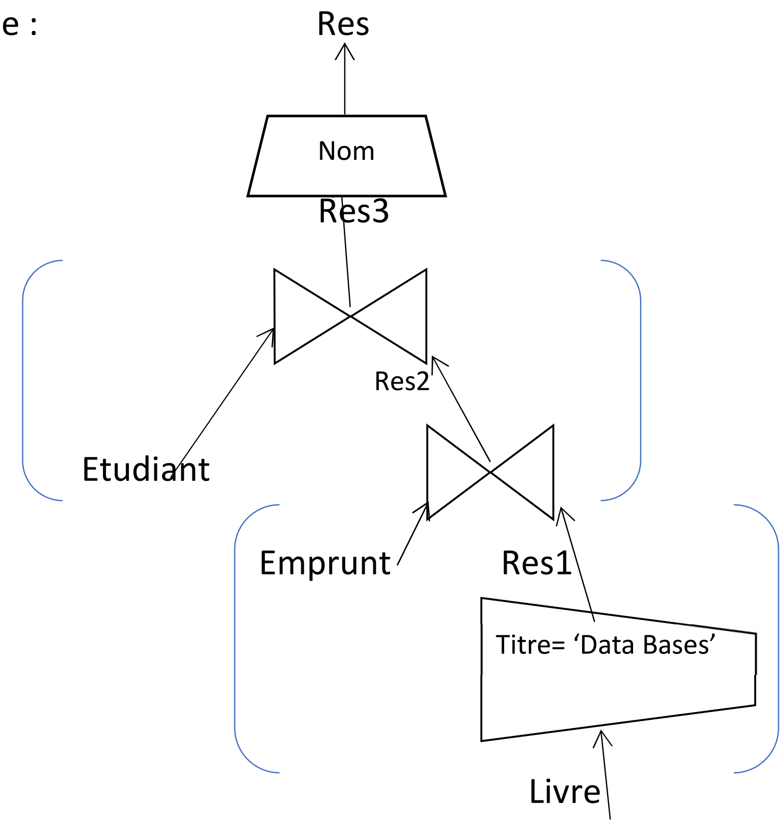
1- 1_ Titres des livres écrits par l'auteur 'Date' . Opération : SELECTION

RES := $\sigma_{Auteur='Date'}$(LIVRE)

2- Nom des étudiants qui ont emprunté le livre 'Data Bases '. Opération de base : JOINTURE

Res := Π_{Nom} (ETUDIANT $\bowtie$ (EMPRUNT $\bowtie$ ($\sigma_{Titre='Data\ Bases'}$(LIVRE))))

Représentation Graphique :



Population de chaque résultat intermédiaire.

Code	Titre	Auteur	ISBN
1A	Data Bases	Date	13254

Res1= Résultat de la sélection

Code	**Matricule**	Date_Emprunt	Date_Retour	Titre	Auteur	ISBN
1A	231_IN	10/10/2019	25/10.2019	Data Bases	Date	13254
1A	342_MI	10/10/2019	25/10.2019	Data bases	Date	13254
1A	111_IN	04/01/2020	19/01/2020	Data Bases	Date	13254

Res2= Résultat de la jointure Emprunt, Res1

Matricule	Nom	Date_Naiq	**Code**	Date_Em-prunt	Date_Re-tour	Titre	Auteur	ISBN
111_IN	Ben Ali	01/02/2000	1A	10/10/2019	25/10.2019	Data Bases	Date	13254
342_MI	Amiar	12/12/1999	1A	10/10/2019	25/10.2019	Data bases	Date	13254
231_IN	Salmi	03/06/2001	1A	04/01/2020	19/01/2020	Data Bases	Date	13254

Res3= Résultat de la Jointure ETUDIANT, Res2

Res	Nom
	Ben Ali
	Amiar
	Salmi

La population du résultat final

- On peut aussi ne garder que les attributs de jointure entre les relations (avant chaque opération de jointure) ; ce qui donne :

$$Res := \Pi_{Nom} (\Pi_{Nom,matricule} (ETUDIANT) \bowtie (\Pi_{matricule,code} (EMPRUNT) \bowtie (\Pi_{code}(\sigma_{Titre='Data\ Bases'}(LIVRE)))))$$

3- Livres qui n'ont jamais été empruntés : ce sont les livres dont le code existe dans la table LIVRE et n'existe pas dans la table EMPRUNT. (Opération : Différence

$$Res := \Pi_{Code} (LIVRE) - \Pi_{Code} (EMPRUNT)$$

4- Livres qui sont empruntés par tous les étudiants. Ce sont les livres dont le code existe dans la table EMPRUNT associé au matricule chaque étudiant (Le même matricule emprunté par tous les étudiants). Opération principale : Division

$$Res := \Pi_{Code,Matricule} \ / \ \underbrace{\Pi_{Matricule} (ETUDIANT)}_{\text{Tous les étudiants}}$$

La relation Res contient une seule colonne (Code) sans aucune ligne.