Chapitre 5
Algèbre relationnel

I. Introduction

Les langages tels que SQL reposent sur l'algèbre relationnelle. L'**algèbre relationnelle** est un langage algébrique qui utilise un ensemble d'opérateurs relationnels combinés pour construire des opérations sur une ou plusieurs tables. Le **langage algébrique** est un langage formel qui offre une base théorique solide au modèle relationnel. Le résultat d'une opération est une table.

Codd a initialement introduit huit opérations (implémentées par le langage SQL), dont certaines peuvent être composées à partir d'autres. Ces opérations peuvent être regroupées en deux types d'opérateurs :

- Les opérateurs relationnels ensemblistes :
  - union
  - intersection
  - différence
  - produit cartésien
- Les opérateurs relationnels spécifiques :
  - restriction ou sélection
  - projection
  - jointure
  - division
  - renommage ou changement de nom

Les opérateurs sont de deux types :

- **unaire** : opère sur une relation
- **binaire** : opère sur deux relations

Le tableau suivant récapitule les opérateurs relationnels dans les différentes catégories présentées précédemment :

OPERATEURS	UNAIRES	BINAIRES
ENSEMBLISTES		union intersection différence produit cartésien
SPECIFIQUES	restriction ou sélection projection renommage	jointure division

Les cinq opérations fondamentales sont : **l'union**, **l'intersection**, **la selection**, **la projection**, **la jointure** et **le renommage**. Les autres opérations peuvent se déduire aisément à partir de ces opérations fondamentales. C'est la raison pour laquelle, certains SGBD n'implémentent pas toutes ces opérations mais n'utilisent que les opérations fondamentales pour répondre à tout type de requête.

II. Les opérateurs ensemblistes

Les opérateurs suivants sont des opérateurs **binaires** : **ils opèrent sur deux relations**.

1. Union

Définition :
L'**union** est une opération portant sur **deux relations de même schéma** R1 et R2. Elle consiste à construire une relation RESULT de même schéma que R1 et R2 ayant pour tuples ceux appartenant à R1 **ou** R2 ou aux deux relations.

Notation : RESULT = UNION (R1,R2) = R1 ∪ R2

Remarques :
L'union ne peut porter que sur deux relations ayant le **même schéma de table** c'est-à-dire sur deux tables ayant strictement les **mêmes attributs**. On dit dans ce cas que les relations sont **compatibles**.

Exemple :

Relation R1

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	20 000
1952LM62	Rover	Bleue	5	28 000
3664PN75	Citroën	Rouge	6	30 000

Relation R2

Immatriculation	Marque	Couleur	Puissance	Prix
3664PN75	Citroën	Rouge	6	30 000
3671KH60	Peugeot	Verte	6	10 000
4691TR95	Renault	Jaune	7	35 000

La relation RESULT renvoie les tuples présents dans R1 ou R2, ou dans les deux.

Relation RESULT

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	20 000
1952LM62	Rover	Bleue	5	28 000
3664PN75	Citroën	Rouge	6	30 000
3671KH60	Peugeot	Verte	6	10 000
4691TR95	Renault	Jaune	7	35 000

2. *Intersection*

Définition :
L'**intersection** est une opération portant sur **deux relations de même schéma** R1 et R2. Elle consiste à construire une relation RESULT de même schéma que R1 et R2 ayant pour tuples ceux appartenant **à la fois** à R1 **et** R2.

Notation : RESULT = INTERSECT (R1,R2) = AND (R1, R2) = R1 $\cap$ R2

Remarques :
L'intersection ne peut porter que sur deux relations ayant le **même schéma de table** c'est-à-dire sur deux tables ayant strictement les **mêmes attributs**.

Exemple :

Relation R1

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	20 000
1952LM62	Rover	Bleue	5	28 000
3664PN75	Citroën	Rouge	6	30 000

Relation R2

Immatriculation	Marque	Couleur	Puissance	Prix
3664PN75	Citroën	Rouge	6	30 000
3671KH60	Peugeot	Verte	6	10 000
4691TR95	Renault	Jaune	7	35 000

La relation RESULT renvoie les tuples présents à la fois dans R1 et dans R2.

Relation RESULT

Immatriculation	Marque	Couleur	Puissance	Prix
3664PN75	Citroën	Rouge	6	30 000

3. *Différence*

Définition :
La **différence** est une opération portant sur **deux relations de même schéma** R1 et R2. Elle consiste à construire une relation RESULT de même schéma que R1 et R2 ayant pour tuples ceux **appartenant à** R1 et **n'appartenant pas à** R2.

Notation : RESULT = MINUS (R1,R2) = R1 – R2 = R1 \ R2

Remarques :
La différence n'est pas commutative, ce qui signifie que R1 – R2 est différent de R2 – R1. La différence ne peut porter que sur deux relations ayant le **même schéma de table** c'est-à-dire sur deux tables ayant strictement les **mêmes attributs**.

Exemple :

Relation R1

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	20 000
1952LM62	Rover	Bleue	5	28 000
3664PN75	Citroën	Rouge	6	30 000

Relation R2

Immatriculation	Marque	Couleur	Puissance	Prix
3664PN75	Citroën	Rouge	6	30 000
3671KH60	Peugeot	Verte	6	10 000
4691TR95	Renault	Jaune	7	35 000

La relation RESULT renvoie tous les tuples appartenant à R1 mais pas à R2.

Relation RESULT

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	20 000
1952LM62	Rover	Bleue	5	28 000

4. *Produit*

Définition :
Le **produit cartésien** de deux relations R1 et R2 (de **schéma quelconque**) consiste à construire une relation RESULT ayant pour schéma la **concaténation des attributs** des deux relations et pour tuples **toutes les combinaisons possibles des tuples** des relations opérandes.

Notation : RESULT = PRODUCT (R1, R2) = R1 $\times$ R2

Remarques :
La relation RESULT comporte autant de lignes que le produit du nombre de lignes de R1 par le nombre de lignes de R2. Les deux relations n'ont pas besoin d'avoir le même schéma de relation.

Exemple :

Relation R1

Immatriculation	Marque	Couleur	Puissance
4578QS59	Renault	Noire	7
1952LM62	Rover	Bleue	5
3664PN75	Citroën	Rouge	6

Relation R2

Prix
30 000
10 000

La relation RESULT renvoie la concaténation de chaque tuple de R1 avec chaque tuple de R2.

Relation RESULT

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	30 000
1952LM62	Rover	Bleue	5	30 000
3664PN75	Citroën	Rouge	6	30 000
4578QS59	Renault	Noire	7	10 000
1952LM62	Rover	Bleue	5	10 000
3664PN75	Citroën	Rouge	6	10 000

III. Les opérateurs relationnels spécifiques

1. *Restriction*

Définition :
La restriction, encore appelée sélection, est une opération unaire. Elle consiste à sélectionner un ensemble de lignes d'une relation en fonction d'un critère de sélection (prédicat ou expression logique de prédicats). Le résultat d'une restriction est une relation de même schéma que la relation initiale.

Notation : RESULT = RESTRICT (R1, Attribut(s) = Valeur(s)) = $\sigma_{\text{Attribut(s) = Valeur(s)}}$ (R1)

Exemple :

Relation R1

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	20 000
1952LM62	Rover	Bleue	5	28 000
3664PN75	Citroën	Rouge	6	30 000
3671KH60	Peugeot	Verte	6	10 000
4691TR95	Renault	Jaune	7	35 000

La relation RESULT = $\sigma_{\text{Marque='Renault'}}$ (R1) renvoie la restriction de R1 sur le prédicat Marque = 'Renault'.

Relation RESULT

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	20 000
4691TR95	Renault	Jaune	7	35 000

2. *Projection*

Définition :
La projection est une opération unaire qui sélectionne un ensemble de colonnes d'une relation, en éliminant les tuples identiques, c'est à dire supprime les tuples (lignes) ayant le même ensemble de valeurs (doublons).

Notation : RESULT = PROJECT (R1, Attribut(s)) = $\Pi_{\text{Attribut(s)}}$ (R1)

Exemple :

Relation R1

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	20 000
1952LM62	Rover	Bleue	5	28 000
3664PN75	Citroën	Rouge	6	30 000
3671KH60	Peugeot	Verte	6	10 000
4691TR95	Renault	Jaune	7	35 000

La relation RESULT = $\Pi_{\text{Marque,Puissance}}$ (R1) renvoie la projection de R1 sur les attributs Marque et Puissance.

Relation RESULT

Marque	Puissance
Renault	7
Rover	5
Citroën	6
Peugeot	6

3. *Jointure*

La jointure est une des opérations essentielles de l'algèbre relationnelle. La jointure permet de composer deux relations à l'aide d'une condition de rapprochement des tables appelée **critère de jointure**. Elle peut être vue comme une restriction du produit cartésien par une condition permettant de comparer des attributs.

Définition :
La **jointure** est une opération binaire. Elle consiste à rapprocher (selon une condition) les tuples de deux relations R1 et R2 afin de former une troisième relation RESULT. Cette relation RESULT contient l'ensemble de tous les tuples obtenus en concaténant un tuple de R1 et un tuple de R2 qui vérifient la condition de rapprochement.

Lorsque le critère de restriction est l'égalité, on parle d'**équijointure**, sinon on parle de **theta-jointure**. La condition de rapprochement des tables doit porter sur des colonnes compatibles. La condition de rapprochement ne porte pas nécessairement sur les attributs clés.

Notation : RESULT = JOIN (R1, R2, R1.attribut = R2.attribut) = R1 ⋈ R2

Exemple :

Relation R1

Immatriculation	Marque	Couleur	Puissance	Prix
4578QS59	Renault	Noire	7	20 000
1952LM62	Rover	Bleue	5	28 000
3664PN75	Citroën	Rouge	6	30 000
3671KH60	Peugeot	Verte	6	10 000
4691TR95	Renault	Jaune	7	35 000

Relation R2

Immatriculation	numPersonne	dateAchat
4578QS59	1591259	12/12/2001
1952LM62	2630662	08/11/1998
4691TR95	1120606	05/09/1997

La relation RESULT = = R1 ⋈ R2 renvoie la jointure de R1 et R2 portant sur la clé Immatriculation.

Relation RESULT

Immatriculation	Marque	Couleur	Puissance	Prix	numPersonne	dateAchat
4578QS59	Renault	Noire	7	20 000	1591259	12/12/2001
1952LM62	Rover	Bleue	5	28 000	2630662	08/11/1998
4691TR95	Renault	Jaune	7	35 000	1120606	05/09/1997

4. 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 simple.

Définition :

La **division** est une opération consistant à construire le quotient de la relation R1 ($A_1,\ldots,A_p, A_{p+1},\ldots,A_n$) par la relation R2 ($A_{p+1},\ldots,A_n$). Le résultat de la division est la relation RESULT ($A_1,\ldots,A_p$) dont les tuples sont ceux qui, concaténés à tout tuple de R2, donnent un tuple de R1.

Notation : RESULT = DIVISION (R1, R2) = R1 / R2 = R1 ÷ R2

Remarques :

Il est à noter que si R1 possède N1 tuples et R2 en possède N2, alors la division de R1 par R2 possédera au plus N1 / N2 tuples. La division n'est pas commutative, ce qui signifie que R1 / R2 est différent de R2 / R1.

Exemple :

Relation R1

Marque	Couleur	Puissance	Prix
Renault	Noire	7	20 000
Rover	Bleue	5	28 000
Citroën	Rouge	6	30 000
Peugeot	Verte	6	10 000
Renault	Jaune	7	35 000

Relation R2

Prix
30 000
10 000

La relation RESULT renvoie la division de R1 par R2, c'est-à-dire que ceci correspond à une requête qui détermine toutes les marques de voiture dont le prix est soit 10 000, soit 30 000.

Relation RESULT

Marque	Couleur	Puissance
Citroën	Rouge	6
Peugeot	Verte	6

5. *Renommage*

Le renommage ou le changement de nom permet de renommer différemment un attribut ou une liste d'attributs d'une relation afin de pouvoir réaliser plus aisément les huit opérations précédentes.

Définition :
Le **renommage** est une opération unaire consistant à renommer un ou plusieurs attributs $A_{p+1},\ldots,A_n$ de la relation R1 $(A_1,\ldots,A_p,\ A_{p+1},\ldots,A_n)$ en $B_{p+1},\ldots,B_n$. Le résultat du renommage est la relation $RESULT(A_1,\ldots A_n,\ B_{p+1},\ldots,B_n)$.

Notation : $RESULT = RENAME\ (R1,\ (A_{p+1},\ldots,A_n)\rightarrow(\ B_{p+1},\ldots,B_n)) = \rho_{\ Attribut(s)\ \ A\ :\ Attribut(s)\ \ B}\ (R1)$

$= \delta_{\ Attribut(s)\ \ A\ :\ Attribut(s)\ \ B}\ (R1)$

Exemple :

Relation R1

Marque	Couleur	Puissance	Prix
Renault	Noire	7	20 000
Rover	Bleue	5	28 000
Citroën	Rouge	6	30 000
Peugeot	Verte	6	10 000
Renault	Jaune	7	35 000

La relation RESULT renvoie le renommage de la colonne « Puissance » par « CV » de R1.

Relation RESULT

Marque	Couleur	CV	Prix
Renault	Noire	7	20 000
Rover	Bleue	5	28 000
Citroën	Rouge	6	30 000
Peugeot	Verte	6	10 000
Renault	Jaune	7	35 000

IV. Définitions ensemblistes des opérateurs algébriques

Pour chaque opération algébrique des paragraphes précédents, on retrouve la définition ensembliste parmi la liste suivante :

– **Union** : $r \cup s = \{\ t \mid [(t \in r) \text{ ou } (t \in s)] \text{ et } (R = S)\ \}$

– **Intersection** : $r \cap s = \{\ t \mid [(t \in r) \text{ et } (t \in s)] \text{ et } (R = S)\ \}$

– **Différence** : $r - s = r \setminus s = \{\ t \mid [(t \in r) \text{ et } (t \notin s)] \text{ et } (R = S)\ \}$

– **Produit cartésien** : $r \times s = \{\ t[RS] \mid [(t[R] \in r) \text{ et } (t[S] \in s)] \text{ et } (R \cap S = \varnothing)\ \}$

– **Restriction** : $\sigma_{\ F}\ (r) = \{\ t \in r \mid r \text{ est satisfaite par la condition } F\ \}$

ex : $\sigma_{\ Y\ =\ y}\ (r) = \{\ t \in r \mid (t[Y] = y) \text{ et } (Y \subseteq R)\ \}$

– **Projection** : $\Pi_{\ X}\ (r) = \{\ t[Y] \mid (t \in r) \text{ et } (Y = X\ \cap R)\ \}$

– **Jointure** : $r \bowtie s = \{\ t[RS] \mid [(t[R] \in r) \text{ et } (t[S] \in s)] \text{ et } (R \cap S \neq \varnothing)\ \}$

– **Division** : $r\ /\ s = r \div s = \{\ t[Y] \mid [(t \in r) \text{ et } (t[S] \in s)] \text{ et } (Y = R - S)\ \}$

– **Renommage** : $\rho_{\ Y\ :\ X}\ (r) = \delta_{\ Y\ :\ X}\ (r) = \{\ t \in r \mid \text{on renomme } Y \text{ par } X\ \}$

Où r et s sont deux relations définies respectivement sur les schémas de relation R et S ; X et Y sont un ensemble de colonnes, y est la valeur d'un n-uplet défini sur Y.

Remarque :
On note l'union entre deux schémas de relation R et S par RS = R ∪ S, et on note la restriction d'un tuple à X par t[X].

V. Quelques propriétés ensemblistes

Voici quelques propriétés qui peuvent être démontrées à l'aide des définitions ensemblistes précédentes :

– **Opérateurs ensemblistes :**
1. $(r \cup s) \cup z = r \cup (s \cup z) = r \cup s \cup z$.
2. $(r \cap s) \cap z = r \cap (s \cap z) = r \cap s \cap z$.
3. $(r \cup s) \cap z = (r \cup z) \cap (s \cup z)$
4. $(r \cap s) \cup z = (r \cap z) \cup (s \cap z)$

– **Sélection avec conditions simples ou complexes :**
1. $\sigma_F (r) = \sigma_F [\sigma_F (r)]$.
2. $\sigma_{F1 \cap F2} (r) = \sigma_{F1} [\sigma_{F2} (r)] = \sigma_{F2} [\sigma_{F1} (r)]$
3. $\sigma_{F1 \cup F2} (r) = \sigma_{F1} (r) \cup \sigma_{F2} (r)$.

– **Projection :**
1. Pour tout X et Y tels que $Y \subseteq X \subseteq R$: $\Pi_Y [\Pi_X (r)] = \Pi_X (r)$

– **Sélection et opérateurs ensemblistes (pour les trois propriétés ci-dessous, on a R = S) :**
1. $\sigma_F (r - s) = \sigma_F (r) - \sigma_F (s) = \sigma_F (r) - s$
2. $\sigma_F (r \cup s) = \sigma_F (r) \cup \sigma_F (s)$
3. $\sigma_F (r \cap s) = \sigma_F (r) \cap \sigma_F (s) = \sigma_F (r) \cap s = r \cap \sigma_F (s)$

– **Projection et opérateurs ensemblistes (pour les trois propriétés ci-dessous, on a R = S) :**
1. $\Pi_X (r \cup s) = \Pi_X (r) \cup \Pi_X (s)$
2. $\Pi_X (r \cap s) \subseteq \Pi_X (r) \cap \Pi_X (s)$
3. $\Pi_X (r) - \Pi_X (s) \subseteq \Pi_X (r - s)$

– **Jointure et opérateurs ensemblistes :**
1. Si R = S, alors $r \bowtie s = r \cap s$
2. Si Q = R, alors $(q \cup r) \bowtie s = (q \bowtie s) \cup (r \bowtie s)$
3. Si Q = R, alors $(q \cap r) \bowtie s = (q \bowtie s) \cap (r \bowtie s)$
4. Si Q = R, alors $(q - r) \bowtie s = (q \bowtie s) - (r \bowtie s)$

– **Produit cartésien et opérateurs ensemblistes (pour les trois propriétés ci-dessous, on a Q = R) :**

5. $(q \cup r) \times s = (q \times s) \cup (r \times s)$
6. $(q \cap r) \times s = (q \times s) \cap (r \times s)$
7. $(q - r) \times s = (q \times s) - (r \times s)$

– **Règles de commutativité :**
1. $\sigma_{F1} [\sigma_{F2} (r)] = \sigma_{F2} [\sigma_{F1} (r)]$.
2. $r \cup s = s \cup r$
3. $r \cap s = s \cap r$
4. $r \times s = s \times r$
5. $r \bowtie s = s \bowtie r$

– **Règles d'associativité :**
1. $(r \bowtie s) \bowtie z = r \bowtie (s \bowtie z)$ si $[(R \cap S = \varnothing)$ et $(Z \cap S = \varnothing)]$ ou $[(R \cap S \neq \varnothing)$ et $(Z \cap S \neq \varnothing)]$
2. $(r \times s) \times z = r \times (s \times z)$ si $[(R \cap S = \varnothing)$ et $(Z \cap S = \varnothing)]$

- **Jointure, sélection, projection et produit cartésien :**
  1. Pour tout R et S tel que $R \subseteq Q$ et $S \subseteq Q$ et $R \cap S \neq \varnothing$, on a : $\Pi_{RS}(q) \subseteq \Pi_R(q) \bowtie \Pi_S(q)$
  2. $\sigma_F (r \bowtie s) = \sigma_F(r) \bowtie \sigma_F(s)$
  3. $\Pi_X (r \bowtie s) = \Pi_X(r) \bowtie \Pi_X(s)$
  4. $\sigma_{F1 \cap F2 \cap F3}(r \times s) = \sigma_{F3}(\sigma_{F1}(r) \times \sigma_{F2}(s))$ si la condition *F1* est définie sur R seulement, la condition *F2* sur S seulement, et la *condition F3* sur R et S.
- **Réécriture des opérateurs déduits :**
  1. <u>Intersection :</u>
     Soit deux schémas de relation R et S tels que R=S, et deux relations r et s définies respectivement sur R et S. Alors on peut écrire l'opération d'intersection par :
     $r \cap s = r - (r - s) = s - (s - r)$
     $r \cap s = (r \cup s) - [(r - s) \cup (s - r)]$
  2. <u>Jointure :</u>
     Soit deux schémas de relation R et S tels que $R \cap S \neq \varnothing$, et deux relations r et s définies respectivement sur R et S. On note R = XY et S = YZ. Alors on peut écrire l'opération de jointure naturelle par :
     $r \bowtie s = \Pi_{RS}[\sigma_{Y = Y'}(r \times \delta_{Y:X}(s))] = \Pi_{RS}[\sigma_{Y = Y'}(s \times \delta_{Y:X}(r))]$
  3. <u>Division :</u>
     Soit deux schémas de relation R et S tels que $S \subseteq R$, et deux relations r et s définies respectivement sur R et S. Alors on peut écrire l'opération de division par :
     $r / s = \Pi_{R-S}(r) - \Pi_{R-S}[(\Pi_{R-S}(r) \times s) - r]$