# Chapitre V : Le calcul relationnel

**Introduction :** Le calcul relationnel est un langage non procédural basé sur le calcul de prédicats du premier ordre qui est un langage logique possédant une syntaxe et une sémantique formelles. Le calcul relationnel permet de dire ce que l'on veut obtenir mais pas comment l'obtenir.

## I. Calcul relationnel à variable n_uplet

### I. 1. Définition

Une requête en calcul relationnel à variable n_uplet est telle que les variables qui y figurent sont des enregistrements. Elle s'exprime comme suit :

**{t / P(t)} :** l'ensemble des n_uplets **t** tels que le prédicat **P(t)** est vrai.

### I. 2. Notion de prédicat

Un prédicat **P** est une expression qui peut comporter des paramètres et qui peut être évaluée à Vrai ou faux (condition booléenne). Les prédicats sont de la forme :

- $r_1(t) \Leftrightarrow t \in r_1$ où $r_1$ est l'instance d'une relation R donnée ;
- t.A Θ Valeur ; // t est un n_uplet (enregistrement) et A est un attribut de la relation R
- $r_1(t_1)$, $r_1(t_2)$ / $t_1$.Attribut_n Θ $t_2$.Attribut_m ; // Θ est un opérateur de comparaison
- Toute combinaison de ces prédicats est un prédicat.

**Exemple**

Soit la relation **Etudiant** (<u>Matricule</u>, Nom, Prénom, Age, Adresse, VilleNaiss) d'instance **e**. Les requêtes suivantes sont écrites en calcul relationnel comme suit :

1. Quels sont les étudiants dont le Nom de famille est "SAMB" ?

   {t / e(t) ∧ (t.Nom = 'SAMB')}

2. Quels sont les étudiants qui habitent à Dakar ?

   {t / e(t) ∧ (t.Adresse = 'Dakar')}

3. Nom des étudiants habitant "Ziguinchor" ou "Bignona" et ayant 20 ans.

   {t.Nom / e(t) ∧ ((t.Ville = 'Ziguinchor') ∨ (t.Ville='Bignona')) ∧ (t.Age = 20)}

### I. 3. Notion de quantificateur

Il existe deux quantificateurs permettant de lier les variables :

### I. 3. 1. Le quantificateurs $\exists$ (il existe)

$\exists$ **t / P(t)** est vrai s'il existe un n_uplet **t** dans la base qui vérifie le prédicat **P(t)**.

**Exemple**

Soit la relation **Salle** (NumSalle, NumBat, Capacité, AnnéeConst) ayant pour instance **s**.

Quels sont les numéros des bâtiments construits la même année ?

$\{t.NumBat\ /\ s(t) \wedge \exists t_1 \in s \wedge (t_1.AnnéeConst = t.AnnéeConst) \wedge (t_1.NumBat \neq t.NumBat)\}$

### I. 3. 2. Le quantificateur $\forall$ (quelque soit)

$\forall$ t P(t) signifie que pour tous les n_uplets le prédicat P(t) est vrai.

**Exemple**

On reprend la relation de l'exemple précédent :

Quels sont les numéros des bâtiments dont toutes les salles ont la même capacité ?

$\{t.NumBat\ /\ s(t) \wedge \forall t_1 \in s, (t_1.Capacité = t.Capacité)\}$

### I. 4. Formalisme

Une expression en calcul relationnel à variable n_uplet est de la forme : **{t / P(t)}**

Dans cette expression, P est une formule et t est un n_uplet. Plusieurs variables n_uplets peuvent apparaître dans une même formule.

### I. 4. 1. Formule

**a) Formule atomique :** Une formule est atomique si elle permet de comparer deux valeurs d'attribut ou une valeur d'attribut et une constante.

Les opérateurs utilisés sont : =, <, >, ≤, ≥, ≠.

**Exemple**

Avec toujours la même relation Salle

- {x, y / x.Capacité = y.Capacité},
- {z / z.NumSalle = 5}

**b) Formule complexe :** Une formule complexe est une combinaison de formules atomiques reliées par des connecteurs ($\vee$ ,$\wedge$, $\leftrightarrow$) et des quantificateurs ( $\exists$ , $\forall$ ).

**Remarque :** Soit P et Q des formules alors :

- $\neg$P, P $\wedge$ Q, P $\vee$ Q sont des formules ;
- $\exists$ r (P(r)) est une formule ;
- $\forall$ r (P(r)) est une formule.

**Remarques :**

- Les quantificateurs $\exists$ r et $\forall$ r permettent de lier la variable r. Une variable non liée est dite variable libre.
- Une restriction importante s'impose à la définition d'une requête {t / P(t)} : La variable t qui apparaît à la gauche du signe "/" doit être la seule variable libre dans la formule P.

Les formules sont constituées à partir d'atomes de la manière suivante :

**a) r(t)** où **t** est une variable n_uplet et **r** une instance relationnelle ;

**b) s.A Θ r.B** où **s** et **r** sont des variables n_uplets, **A** est un attribut de la relation où **s** est définie et **B** est un attribut de la relation où **r** est définie. **Θ** est un opérateur de comparaison.

**c) r.A Θ C** où **C** est une constante, **Θ** un opérateur de comparaison, **r** un n_uplet et **A** un attribut.

### I. 5. Expression des opérations algébriques en calcul relationnel

Toutes les opérations de l'algèbre relationnelle peuvent être exprimées à l'aide des formules du calcul relationnel.

#### I. 5. 1. La sélection

$\sigma_{\Theta}$ **(R)** $\Leftrightarrow$ **{t / r(t)** $\wedge$ **Θ}**

- r est l'instance de la relation R ;
- Θ une condition de sélection ;
- t un n_uplet appartenant à r.

#### I. 5. 2. La projection

$\prod_{A1,A2,\ldots,Aj}$**(R)** $\Leftrightarrow$ **{**$t.A_1, t.A_2, \ldots, t.A_j$ **/ t** $\in$ **r}**

- r est l'instance de la relation R ;
- t un n_uplet ;
- $A_1, A_2, \ldots, A_j$ des attributs de la relation R.

**I. 5. 3. L'union**

$\mathbf{R_1 \vee R_2 \Leftrightarrow \{t \,/\, t \in r_1 \vee t \in r_2\}}$

- $r_1$ et $r_2$ sont respectivement les instances des relations $R_1$ et $R_2$ ;
- t un n_uplet.

**I. 5. 4. La différence**

$\mathbf{R_1 - R_2 \Leftrightarrow \{t \,/\, r_1(t) \wedge \neg(r_2(t))\}}$

- $r_1$ et $r_2$ sont les instances respectives des relations $R_1$ et $R_2$ ;
- t un n_uplet.

**I. 5. 5. Le produit cartésien**

Soient $R_1$ et $R_2$ $_2$ deux relations dont les attributs sont respectivement $A_1, A_2, \ldots, A_n$ et $B_1, B_2, \ldots, B_m$. Le produit cartésien $R_1 * R_2$ avec $r_1$ et $r_2$ les instances respectives de $R_1$ et $R_2$ s'exprime comme suit :

$\mathbf{R_1 * R_2 \Leftrightarrow \{t \,/\, \exists\, u \in r_1 \wedge \exists\, v \in r_2 \wedge (t.A_1 = u.A_1) \wedge \ldots \wedge (t.A_n = u.A_n) \wedge (t.B_1 = v.B_1) \wedge \ldots \wedge (t.B_m = v.B_m)\}.}$

$R_1 * R_2 = \{A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m\}$

**I. 5. 6. La jointure naturelle**

Soient $R_1$ d'attributs $A_1, A_2, \ldots, A_n, C_1, \ldots, C_p$ et $R_2$ d'attributs $B_1, B_2, \ldots, B_m, D_1, \ldots, D_p$. $r_1$ et $r_2$ étant respectivement les instances de $R_1$ et $R_2$.

$\mathbf{R_1\ |><|_{\Theta}\ R_2 \Leftrightarrow \{t \,/\, \exists\ u, \exists\ v \wedge r_1(u) \wedge r_2(v) \wedge (t.A_1 = u.A_1) \wedge \ldots \wedge (t.A_n = u.A_n) \wedge (t.B_1 = v.B_1) \wedge \ldots \wedge (t.B_m = v.B_m) \wedge (u.C_1 = v.D_1) \wedge (t.C_1 = u.C_1) \wedge (t.D_1 = v.D_1) \wedge \ldots \wedge (u.C_p = v.D_p) \wedge (t.C_p = u.C_p) \wedge (t.D_p = v.D_p)\}}$

$R_1\ |><|_{\Theta}\ R_2 = \{A_1, A_2, \ldots, A_n, C_1, \ldots, C_p, B_1, B_2, \ldots, B_m, D_1, \ldots, D_p\}.$

$\Theta$ est la condition de jointure telle que : $\Theta = ((C_1 = D_1) \wedge (C_2 = D_2) \wedge \ldots \wedge (C_p = D_p))$

**Exemple :** Soient les relations suivantes :

**Etudiant** (<u>NumCarte</u>, Nom, Prénom, Adresse, Filière, Age) d'instance **e**

**Cours** (<u>Code</u>, Libellé, VolHoraire) d'instance **c**

**Suivre** (<u>#Etudiant</u>, <u>#Cours</u>) d'instance **s**

**Questions**

1. Donnez les noms des étudiants âgés de 24 ans et qui habitent Diourbel ou Ziguinchor.

2. Donnez l'adresse et le nom de chaque étudiant qui suit le cours de base de données.
3. Quels sont les cours dont le volume horaire dépasse 24h ?
4. Quels sont les libellés des cours suivis par l'étudiant Moustapha SAMB ?
5. Quels sont les cours qui ont le plus grand volume horaire ?

**Réponses**

1. {t.Nom / e(t) ∧ (t.Age = 24) ∧ ((t.Adresse = 'Diourbel') ∨ (t.Adresse = 'Ziguinchor'))}
2. {t.Nom, t.Adresse / e(t) ∧ ∃ $t_1$ ∈ s, ∃ $t_2$ ∈ c ∧ (t.NumCarte = $t_1$.NumCarte) ∧ ($t_1$.Code = $t_2$.Code) ∧ ($t_2$.Libellé = 'Base de données')}
3. {t / c(t) ∧ (t.Volume > 24)}
4. {t.Libellé / c(t) ∧ ∃ $t_1$ ∈ e ∧ ∃ $t_2$ ∈ s ∧ (t.Code = $t_2$.Code) ∧ (t.NumCarte = $t_1$.NumCarte) ∧ ($t_1$.Nom = 'SAMB') ∧ ($t_1$.Prénom = 'Moustapha')}
5. {t / c(t) ∧ ∀ $t_1$ ∧ c($t_1$) ∧ (t.Volume ≥ $t_1$.Volume}

**Remarque**

De telles expressions peuvent générer un nombre infini de n_uplets. Ainsi, on restreint la définition des formules en introduisant la notion de domaine d'une formule. Le domaine d'une formule P est l'ensemble des valeurs qui y apparaissent explicitement. Le domaine de P est noté dom(P)

## II. Calcul relationnel à variable domaine

A la différence du calcul relationnel à variables n_uplets, les variables portent sur les valeurs d'es attributs. Il utilise donc des variables domaines provenant d'un domaine d'attribut plus que d'une variable n_uplet.

Une expression en calcul relationnel à variable domaine est de la forme :

$$\{<x_1, x_2, \ldots, x_n> / P(<x_1, x_2, \ldots, x_n>)\}$$

1. $x_1, x_2, \ldots, x_n$ représentent des variables domaines ;
2. P est une formule composée d'atomes de la forme :
   - R($x_1, x_2, \ldots, x_n$) où R est une relation de n attributs et $x_1, x_2, \ldots, x_n$ sont des variables domaines ou des constantes domaines ;
   - x **Θ** y où x et y sont des variables domaines et **Θ** un opérateur de comparaison. x et y doivent être des domaines compatibles ;

- $x$ **Θ** $C$ où $C$ est une constante du domaine de l'attribut dont $x$ est une variable domaine.

Une expression est construite à partir de formules telles que :

- Si $P_1$ et $P_2$ sont des formules alors $\neg P_1$, $P_1 \vee P_2$ et $P_1 \wedge P_2$ sont des formules ;
- Si $P(x)$ est une formule alors $\exists\, x\, (P(x))$ et $\forall\, x\, (P(x))$ sont des formules.

**Exemple :** Soient les relations suivantes :

**Etudiant** (<u>NumCarte</u>, Nom, Prénom, Age, Adresse) représentée par $E_1$

**Enseignant** (<u>Matricule</u>, Nom, Prénom, Salaire, Age) représentée par $E_2$

**Questions**

1. Quels sont les étudiants qui habitent à Dakar ?
2. Quels sont les enseignants plus âgés que tous les étudiants ?
3. Quels sont les étudiants plus âgés qu'un de leurs enseignants ?
4. Matricule, Nom et Prénom des enseignants qui gagnent plus de 200000 par mois.
5. Numéro de carte, Age et Adresse des étudiants ayant le même nom qu'un enseignant.

**Réponses**

1. $\{<x_1, x_2, x_3, x_4, x_5> / E_1(<x_1, x_2, x_3, x_4, \text{'Dakar'}>)\}$
2. $\{<x_1, x_2, x_3, x_4, x_5> / E_2(<x_1, x_2, x_3, x_4, x_5>) \wedge \forall E_1(<y_1, y_2, y_3, y_4, y_5>), (x_5 > y_4)\}$
3. $\{<x_1, x_2, x_3, x_4, x_5> / E_1(<x_1, x_2, x_3, x_4, x_5>) \wedge \exists E_2(<z_1, z_2, z_3, z_4, z_5>), (x_4 > z_5)\}$
4. $\{<x_1, x_2, x_3> / E_2(<x_1, x_2, x_3, x_4, x_5>), (x_4 > 200000)\}$
5. $\{<x_1, x_4, x_5> / E_1(<x_1, x_2, x_3, x_4, x_5>) \wedge \exists E_2(<y_1, y_2, y_3, y_4, y_5>), (x_2 = y_2)\}$

## III. Calcul relationnel Vs Algèbre relationnelle

L'algèbre relationnelle et le calcul relationnel ont la même puissance d'expression ; donc toutes les requêtes qui peuvent être formulées en utilisant l'un peuvent aussi l'être en utilisant l'autre. Ce fut vérifié en premier par E.F. Codd en 1972. La preuve est fondée sur un algorithme (appelé « algorithme de réduction de Codd ») par lequel une expression arbitraire du calcul relationnel peut être réduite à une expression au sens équivalent de l'algèbre relationnelle. Certains déclarèrent que les langages basés sur le calcul relationnel sont de « haut niveau » ou « plus déclaratifs » que les langages basés sur l'algèbre relationnelle parce que ce dernier spécifie (partiellement) l'ordre des opérations tandis que le calcul laisse le compilateur ou l'interpréteur déterminer l'ordre d'évaluation le plus efficace.