LICENCES MATHÉMATIQUES ET INFORMATIQUE
3ÈME ANNÉE - FORMATION INITIALE ET PAR APPRENTISSAGE

BASES DE DONNÉES RELATIONNELLES

POLYCOPIÉ DE COURS - CALCUL RELATIONNEL À VARIABLE NUPLET

Maude Manouvrier

Table des matières

Chapitre 4

Calcul relationnel à variable nuplet

Le Calcul Relationnel est un langage non-procédural (déclaratif) : on décrit sous forme logique ce que l'on veut obtenir comme résultat de requête, mais pas comment on l'obtient (par quelles opérations), contrairement à l'Algèbre Relationnelle qui exprime comment on calcule le résultat de la requête (sous la forme d'opérations ensemblistes sur les nuplets).

En calcul relationnel à variable nuplet, on manipule des variables qui représentent des nuplets.

4.1 Définition

Une requête en calcul relationnel à variable nuplet s'exprime de la manière suivante :

$$\{t \mid P(t)\}$$

indiquant que la requête renvoie l'ensemble des nuplets t tels que le prédicats $P(t)$ est vrai.

Ce langage se base sur la logique de prédicats avec comme hypothèses :
— Soit *Tout ce qui est dans la base de données est vraie et tout ce qui n'est pas dans la base est faux.*
— Soit *Tout ce qui est dans la base de données est vraie. Pour le reste, on ne sait pas.* - Cette hypothèse est plus réaliste.

4.1.1 Prédicats

Les prédicats peuvent être de la forme :
- $r_1(t)$ ou $t \in r_1$ qui signifie que t est un nuplet de r_1.
- $t.att_1 = valeur_1$ qui signifie que l'attribut att_1 du nuplet t a pour valeur $valeur_1$, sachant que l'on a précisé avant à quelle relation appartient t.
- $t_1.att_1 > t_2.att_2$ qui compare les attributs att_1 des nuplets t_1 et t_2, en ayant précisé avant de quelles relations sont issus t_1 et t_2.
- N'importe quelle combinaison des formules précédentes.

Par exemple, si on veut les étudiants habitant Paris et qui ont plus de deux années d'université :

$$\{t \mid Etudiant(t) \wedge (t.Ville =' Paris') \wedge (t.NumAnnee \geq 2)\}$$

On indique ici que la requête contient l'ensemble des nuplets t de $Etudiant$ tels que $(t.Ville =' Paris')$ et $(t.NumAnnee \geq 2)$.

On peut spécifier les attributs que l'on souhaite dans le résultats :

$$\{t.Nom, t.Prenom \mid Etudiant(t) \wedge (t.Ville =' Paris') \wedge (t.NumAnnee \geq 2)\}$$

On indique ici que la requête contient l'ensemble des attributs Nom et $Prenom$ des nuplets t de $Etudiant$ tels que $(t.Ville =' Paris')$ et $(t.NumAnnee \geq 2)$.

4.1.2 Quantificateurs

On peut utiliser les quantificateurs $\forall$ et $\exists$.

- $\exists\ t\ (P(t))$ vraie signifie qu'il existe un nuplet t dans la base de données qui vérifie le prédicat $P(t)$.

  Par exemple si l'on veut les étudiants de même nom qu'un enseignant :
  $\{t.Nom, t.Prenom \mid Etudiant(t) \wedge\ (\ \exists s\ (Enseignant(s)\ ) \wedge (s.Nom = t.Nom)\ )\ \}$
  On indique ici que la requête contient l'ensemble des nuplets t de $Etudiant$ tels qu'il existe un nuplet s dans $Enseignant$ avec mes mêmes valeurs que t sur les attributs Nom et $Prenom$.

- $\forall\ t\ (P(t))$ signifie que pour tous les nuplets de la base, $P(t)$ est vrai avec $P(t)$ de la forme $A(t) \Longrightarrow B(t)$.

  Attention : la requête avec un $\forall$ doit être bien formée.

  Par exemple, la requête $\forall t\ (Etudiant(t) \wedge (t.Ville =' Paris')\ )$ n'est pas bien formée car elle signifie que tous les nuplets de la base de données sont des nuplets de $Etudiant$ et sont tels que $(t.Ville =' Paris')$.

  En revanche, si la base de données ne contient dans la relation $Etudiant$ que des nuplets correspondant à des étudiants parisiens, alors la requête suivante est bien formée :
  $\forall t\ Etudiant(t) \Longrightarrow (t.Ville =' Paris')$ car elle signifie que pour tous les nuplets de la base, **s'il s'agit** de nuplets de $Etudiant$ **alors** ils vérifient $(t.Ville =' Paris')$.

Une variable quantifiée (i.e. précédée d'un $\exists$ ou d'un $\forall$) est dite **variable liée**, sinon il s'agit d'une **variable libre**.

4.1.3 Équivalence logique et expression saine

Pour rappel, en logique on a les équivalences suivantes :

- $P_1 \wedge P_2$ est équivalent à $\neg[\neg(P_1) \vee \neg(P_2)]$

- $\forall t\ P(t)$ est équivalent à $\neg(\exists t\ (\neg P(t)))$

Il faut faire attention car de telles expressions peuvent générer un nombre infini de nuplets et donc être non saines. Une expression saine en calcul relationnel à variable nuplet peut s'exprimer par une requête équivalente (i.e. donnant le même résultat) en algèbre relationnelle.
Par exemple, l'expression $\{t \mid (\neg Etudiant(t))\}$ n'est pas saine car cela signifie que l'on veut tous les nuplets qui n'appartiennent pas à la relation $Etudiant$.

4.2 Expression des opérateurs algébriques en calcul relationnel à variable nuplet

4.2.1 Sélection

$\sigma_\theta(r)$ s'exprime en en calcul relationnel à variable nuplet par : $\{t \mid r(t) \wedge \theta(t)\}$
Par exemple, si on veut les étudiants habitant Paris et qui ont plus de deux années d'université :

$$\{t \mid Etudiant(t) \wedge (t.Ville =' Paris') \wedge (t.NumAnnee \geq 2)\}$$

4.2.2 Projection

$\Pi_{A_1,A_2,...,A_n}(r)$ s'exprime en en calcul relationnel à variable nuplet par :

$$\{t.A_1, t.A_2, ..., t.A_n \mid r(t)\}$$

Par exemple, si on veut les noms et prénoms des étudiants : $\{t.Nom, t.Prenom \mid Etudiant(t)\}$

4.2.3 Union

$r_1 \cup r_2$ s'exprime en en calcul relationnel à variable nuplet par : $\{t \mid r_1(t) \vee r_2(t)\}$

Par exemple, si on veut les noms et prénoms des enseignants et des étudiants :

$\{t.Nom, t.Prenom \mid Enseignant(t) \vee Etudiant(t)\ \}$

4.2.4 Différence

$R_1 - R_2$ s'exprime en en calcul relationnel à variable nuplet par :

$\{t \mid r_1(t) \wedge \neg[\exists\ u\ s(u) \wedge (t.A_1 = u.B_1) \wedge ... \wedge (t.A_n = u.B_n)]\}$

avec r de schéma $R(A_1, ...A_n)$ et s de schéma $S(B_1, ...B_n)$, schéma-compatibles.

Par exemple, si on veut les noms et prénoms des enseignants qui ne sont pas étudiants :

$\{t.Nom, t.Prenom \mid Enseignant(t) \wedge \neg[\exists\, u\, Etudiant(u) \wedge (t.Nom = u.Nom) \wedge (t.Prenom = u.Prenom)]\}$

On veut dans le résultat de la requête les valeurs des attributs Nom et $Prenom$ des nuplets t de $Enseignant$ tels qu'il n'existe pas un nuplet u dans $Etudiant$ avec les mêmes valeurs que t sur les attributs Nom et $Prenom$.

4.2.5 Produit cartésien

$r * s$, avec r de schéma R ayant pour attributs A_1 à A_n et s de schéma S ayant pour attributs B_1 à B_m, s'exprime en calcul relationnel à variable nuplet par :

$\{t \mid \exists\ u, v\ r(u) \wedge s(v) \wedge (t.A_1 = u.A_1) \wedge ... \wedge (t.A_n = u.A_n) \wedge (t.B_1 = v.B_1) \wedge ... \wedge (t.B_m = v.B_m)\}$

En effet, le nuplet t n'appartient à aucune instance de relation, mais est construit à partir d'un nuplet u de r et d'un nuplet v de s.

On peut également écrire la requête ainsi (ce qui sera plus proche de la requête que l'on écrira en SQL) :

$\{u.A_1, ..., u.A_n, v.B_1, ..., v.B_m \mid r(u) \wedge s(v)\ \}$

4.2.6 Jointure

$r \bowtie s$, avec r de schéma R ayant pour attributs A_1 à A_n et C_1 à C_p et s de schéma S ayant pour attributs B_1 à B_m et C_1 à C_p, la jointure se faisant par exemple sur les attributs de même nom C_1 à C_p, s'exprime en calcul relationnel à variable nuplet par :

$\{t \mid \exists\ u, v\ r(u) \wedge s(v) \wedge (t.A_1 = u.A_1) \wedge ... \wedge (t.A_n = u.A_n) \wedge (t.B_1 = v.B_1) \wedge ... \wedge (t.B_m = v.B_m) \wedge (u.C_1 = v.C_1) \wedge (t.C_1 = u.C_1) \wedge ... \wedge (u.C_p = v.C_p) \wedge (t.C_p = u.C_p)\ \}$

On peut également écrire la requête ainsi (ce qui sera plus proche de la requête que l'on écrira en SQL) :

$\{u.A_1, ..., u.A_n, u.C_1, ..., u.C_p, v.B_1, ..., v.B_m \mid r(u) \wedge s(v) \wedge (u.C_1 = v.C_1) \wedge (t.C_1 = u.C_1) \wedge ... \wedge (u.C_p = v.C_p) \wedge (t.C_p = u.C_p)\ \}$

Par exemple si on veut les noms et prénoms et le nom du département de chaque enseignant :

$\{t.Nom,\ t.Prenom,\ u.NomDepartement \mid Enseignant(t) \wedge\ Department(u) \wedge (t.DepartmentID = u.DepartmentID)]\}$

On indique que le résultat de la requête contient les valeurs des attributs Nom et $Prenom$ des nuplets t de $Enseignant$ et les valeurs de l'attribut $NomDepartement$ de nuplets u de $Department$ tels que t et u aient la même valeur pour l'attribut $DepartmentID$.

4.2.7 Division

Expression en calcul en utilisant $\forall$ et $\Longrightarrow$:

Soient deux relations r, de schéma R ayant pour attributs A_1 à A_n et C_1 à C_p, et s de schéma S ayant pour attributs C_1 à C_p. $r \div s$ signifie que l'on souhaite obtenir les morceaux de nuplets t de r (i.e. de schéma A_1, $...A_n$) tels que pour tous les nuplets u de s, il existe un nuplet v, dans r, ayant même valeur que u pour les attributs C_1 à C_p et même valeur que t pour les attributs A_1 à A_n.

$\{t \mid r(t) \wedge [\forall u\ s(u) \Longrightarrow (\exists v\ r(v) \wedge (u.C_1 = v.C_1) \wedge ... \wedge (u.C_p = v.C_p) \wedge (t.A_1 = v.A_1) \wedge ... \wedge (t.A_n = v.A_n)\ )\ ]\ \}$

Par exemple si on veut les noms et prénoms des enseignants qui interviennent dans les cours de tous les masters :

$\{t.Nom, t.Prenom \mid Enseignant(t) \wedge [\forall u\ Master(u) \Longrightarrow (\exists v\ Cours(v) \wedge (t.EnseignantID = v.EnseignantID) \wedge (u.MasterID = v.MasterID)]\}$

On indique que le résultat de la requête contient les valeurs des attributs Nom et $Prenom$ des nuplets t de $Enseignant$ tels pour tous les nuplets u, si c'est un nuplet de $Master$ alors il existe un nuplet

v dans la relation $Cours$ ayant la mêmes valeur que t sur l'attribut $EnseignantID$ et la mêmes valeur que u sur l'attribut $MasterID$.

Expression en calcul en utilisant $\forall$, $\vee$ et $\neg$:

En logique, l'expression $\forall t\ A(t) \Longrightarrow B(t)$ est équivalente à $\forall t\ \neg(A(t)) \vee B(t)$. En effet, $\forall t\ A(t) \Longrightarrow B(t)$ signifie que pour tous les nuplets t si $A(t)$ est vérifié alors $B(t)$ l'est aussi. $\forall t\ \neg(A(t)) \vee B(t)$ signifie que, pour tous les nuplets t, soit $A(t)$ n'est pas vérifié, soit (donc implicitement $A(t)$ est vérifié) $B(t)$ est vérifié.

On peut donc écrire la division $r \div s$ en calcul relationnel à variable nuplet de la manière suivante :
$\{t \mid r(t) \wedge [\forall u\ \neg(s(u)) \vee (\exists v\ r(v) \wedge (u.C_1 = v.C_1) \wedge ... \wedge (u.C_p = v.C_p) \wedge (t.A_1 = v.A_1) \wedge ... \wedge (t.A_n = v.A_n)\ )\ ]\ \}$

Par exemple si on veut les noms et prénoms des enseignants qui interviennent dans tous les masters :

$\{t.Nom, t.Prenom \mid Enseignant(t) \wedge [\forall u\ \neg(Master(u)) \vee (\exists v\ Cours(v) \wedge (t.EnseignantID = v.EnseignantID) \wedge (u.MasterID = v.MasterID)]\}$

On indique que le résultat de la requête contient les valeurs des attributs Nom et $Prenom$ des nuplets t de $Enseignant$ tels pour tous les nuplets u, soit ce n'est un nuplet de $Master$, soit (u est un nuplet de $Master$ et) il existe un nuplet v dans la relation $Cours$ ayant la mêmes valeur que t sur l'attribut $EnseignantID$ et la mêmes valeur que u sur l'attribut $MasterID$.

4.3 Contraintes

Les contraintes d'intégrité de la base de données peuvent être exprimées en calcul relationnel à variable nuplet. Par exemple, supposons que nous ayons la base de données suivante :

```
Agence(nom_banque,ville ...)
Emprunt(nom_banque,num_client, montant ...)
Compte(nom_banque,num_client,num_compte,solde ...)
```

En calcul relationnel à variable nuplet, la contrainte "*Chaque emprunteur possède un compte en banque dans l'agence dont le solde est au minimum égal à la moitié de son emprunt*" s'exprime de la manière suivante :

$\neg[\exists e\, Emprunt(e) \mid \neg(\exists c\, Compte(c) \wedge (c.num\_client = e.num\_client) \wedge (c.nom\_banque = e.nom\_banque) \wedge (c.solde >= (e.montant \backslash 2))]$
Ce qui signifie qu'il n'est pas possible de trouver un nuplet e dans $Emprunt$ (donc il n'existe pas de nuplet e dans $Emprunt$) pour lequel il n'existe pas de nuplet c dans $Compte$ avec le même identifiant de banque et de client et un montant inférieur au solde.

4.4 Conclusion et points à retenir

- Au lieu de décrire comment réaliser la requête, le calcul relationnel décrit les nuplets de la relation résultat de la requête.
- Les variables du calcul à variable nuplet prennent leurs valeurs dans les nuplets des instances de la base de données.
- Toute requête en algèbre relationnelle peut être exprimée en calcul relationnel.