Calcul relationnel

Skander Zannad et Judicaël Courant

2014-04-04

1 Résumé des épisodes précédents

On a vu

(i) MCD (Entité-Association) pour la représentation conceptuelle d'un problème ;

(ii) MLD pour transcrire le MCD en tables ;

(iii) Implantation dans une base de données SQL (MPD) ;

(iv) Requêtes SQL.

2 Problème

SQL : quelle expressivité ?

Autrement dit : peut-on poser n'importe quel type de requête à SQL ?
Prérequis : modéliser ce problème correctement (mathématiquement).
Modélisation que nous allons utiliser : *le modèle relationnel*.
Deux parties :

Structure du modèle relationnel : données (tables).

Calcul relationnel : les requêtes.

3 Structure du modèle relationnel

Définition 3.0.1 (Attribut, domaine)**.**
Ensemble $\mathbf{att}$ infini : ensemble d'*attributs*.
Pour tout attribut a, $\mathbf{dom}$: ensemble de *constantes* (*type* de a).
$\mathbf{dom}$: union de toutes les constantes de tous les attributs.

Attributs de notre base : `titre`, `nom`, `prenom`, `id`, `date`, ...

Définition 3.0.2 (Schémas)**.**
On suppose donnés :

relname : ensemble de noms de relations (disjoint de $\mathbf{att}$) ;

sort : fonction de $\mathbf{relname}$ dans l'ensemble des ensembles finis d'attributs.

Alors :

Schéma relationnel : élément de $\mathbf{relname}$

Schéma de base de données : ensemble fini de schémas relationnels.

Exemple pour notre base :

$$CINEMA = \{\,\texttt{FILM}, \texttt{JOUE}, \texttt{PERSONNE}, \texttt{PERSONNAGE}\,\}$$
$$\mathbf{sort}(FILM) = \{\,\texttt{id}, \texttt{titre}, \texttt{date}, \texttt{idrealisateur}\,\}$$
$$\mathbf{sort}(PERSONNE) = \{\,\texttt{id}, \texttt{nom}, \texttt{prenom}, \texttt{datenaissance}\,\}$$

Écriture $FILM[\texttt{id titre date idrealisateur}]$ pour dénoter $FILM$ et rappeler $\mathbf{sort}(FILM)$.

Définition 3.0.3 (n-uplets)**.**
Soit U ensemble fini.
nuplet sur U : fonction de U dans $\mathbf{dom}$.
sort du nuplet : U.
Arité du nuplet : $|U|$.
Notation $\langle a_1 : v_1, \dots, a_k : v_k\rangle$.
nuplet sur $\emptyset$: $\langle\,\rangle$.

Exemples :

$$u = \langle \texttt{id} : 1, \texttt{prenom} : "Clint", \texttt{nom} : "Eastwood"\rangle$$

$u(\texttt{prenom}) = "Clint"$.
Notation : $u[\texttt{prenom}, \texttt{nom}] = \langle \texttt{prenom} : "Clint", \texttt{nom} : "Eastwood"\rangle$.
De manière générale : pour u de sort U et $V \subset U$, $u[V] = u_{|V}$.
Généralisation de la notion mathématique :

$$(42, 17, 6) = \langle 1 : 42, 2 : 17, 3 : 6\rangle$$

Définition 3.0.4 (Relation)**.**
Relation sur un ensemble d'attributs U ou *instance d'un schéma relationnel* $R[U]$: ensemble fini de nuplets sur U (à valeurs dans $\mathbf{dom}$).
sort de la relation : U (*sort* de chacun des éléments)
Arité de la relation : $|U|$ (arité de chacun des éléments)

Définition 3.0.5 (Base de données).
I instance d'un schéma de base de données S : fonction définie sur S telle que pour tout $R \in S$, $I(R)$ relation sur R.

Exemple : instances du schéma $CINEMA$: tables données dans les cours précédents.

4 Calcul relationnel

On fixe :

– S, schéma de base de données ;
– I, instance de ce schéma ;
– **var** ensemble infini de noms de variables (disjoint de **dom**).
– Deux valeurs : V et F destinée à représenter le vrai et le faux.

Hypothèse simplificatrice : on suppose **dom** ensemble fini de chaînes de caractères (par exemple toutes celles apparaissant dans la base).

Requête : formule

– portant sur les relations de la base ;
– contenant des variables $x_1, \ldots, x_n$.

Réponse à une requête : ensemble des nuplets u sur $\{ x_1, \ldots, x_n \}$ tels que le remplacement de $x_1, \ldots, x_n$ dans la formule donne une formule vraie.

Plus précisément

Définition 4.0.6 (Termes, Atomes).
Terme t : élément de $\mathbf{var} \cup \mathbf{dom}$.

Formule atomique (ou atome) sur S : expression $R(t_1, \ldots, t_n)$ où n arité de R ou expression de la forme $t_1 = t_2$.

Ensemble des *variables libres* d'une formule atomique F : ensemble des v apparaissant dans F.

$$\begin{aligned} v \in \mathrm{libre}(t_1 = t_2) &\iff (v \in \mathbf{var} \text{ et } (v = t_1 \text{ ou } v = t_2)) \\ v \in \mathrm{libre}(R(t_1, \ldots, t_n)) &\iff (v \in \mathbf{var} \text{ et } \exists i \in \{ 1, \ldots, n \} \ v = t_i) \end{aligned}$$

Assignation de valeur/valuation pour F : nuplet sur $\mathrm{libre}(F)$ (ou un sur-ensemble de $\mathrm{libre}(F)$).

Pour $t \in \mathbf{dom}$ et ν une valuation, on note $\nu(t) = t$.

Définition 4.0.7 (Vérité d'une formule atomique).
Valeur de vérité d'une formule F pour I et une assignation ν :

Notation $[\![F]\!]_{I,\nu}$;

Définition Deux cas :

– $[\![t_1 = t_2]\!]_{I,\nu}$ vaut V si $\nu(t_1) = \nu(t_2)$, F sinon ;
– $[\![R(t_1, \ldots, t_n)]\!]_{I,\nu}$ vaut V si $(\nu(t_1), \ldots, \nu(t_n)) \in I(R)$, F sinon.

Définition 4.0.8 (Formule).
Une *formule* (ou *formule bien formée* sur le schéma S) du calcul relationnel est une expression obtenue par l'application des règles suivantes :

(i) Toute formule atomique sur S est une formule ;
(ii) Pour toutes formules ϕ et ψ, $\phi \wedge \psi$, $\phi \vee \psi$ et $\phi \Rightarrow \psi$ sont des formules ;
(iii) Pour toute formule ϕ, $\neg\phi$ est une formule ;
(iv) Pour toute formule ϕ et toute variable x, $\exists x\ \phi$ et $\forall x\ \psi$ sont des formules.

Définition 4.0.9 (Variables libres d'une formule).
Définie par récurrence sur la taille de la formule :

(i) $\mathrm{libre}(A)$ pour A atomique : déjà défini ;
(ii) $\mathrm{libre}(\phi \wedge \psi) = \mathrm{libre}(\phi) \cup \mathrm{libre}(\psi)$;
(iii) $\mathrm{libre}(\phi \vee \psi) = \mathrm{libre}(\phi) \cup \mathrm{libre}(\psi)$;
(iv) $\mathrm{libre}(\phi \Rightarrow \psi) = \mathrm{libre}(\phi) \cup \mathrm{libre}(\psi)$;
(v) $\mathrm{libre}(\neg\phi) = \mathrm{libre}(\phi)$;
(vi) $\mathrm{libre}(\exists x\ \phi) = \mathrm{libre}(\phi) \setminus \{ x \}$;
(vii) $\mathrm{libre}(\forall x\ \phi) = \mathrm{libre}(\phi) \setminus \{ x \}$.

Définition 4.0.10 (Vérité d'une formule).
Défini par récurrence sur la taille de la formule :

(i) $[\![A]\!]_{I,\nu}$ pour A atomique déjà définie ;
(ii) $[\![\phi \wedge \psi]\!]_{I,\nu}$ vaut V $[\![\phi]\!]_{I,\nu} = V$ et $[\![\phi]\!]_{I,\nu} = V$, vaut F sinon ;
(iii) $[\![\phi \vee \psi]\!]_{I,\nu}$ vaut V si $[\![\phi]\!]_{I,\nu} = V$ ou $[\![\phi]\!]_{I,\nu} = V$, vaut F sinon ;
(iv) $[\![\phi \Rightarrow \psi]\!]_{I,\nu}$ vaut V si $[\![\phi]\!]_{I,\nu} = F$ ou $[\![\phi]\!]_{I,\nu} = V$, F sinon ;
(v) $[\![\neg\phi]\!]_{I,\nu}$ vaut V si $[\![\phi]\!]_{I,\nu} = F$ et F sinon ;

(vi) $[\![\exists x\ \phi]\!]_{I,\nu}$ vaut V s'il existe au moins une valeur $v \in \mathbf{dom}$ telle que $[\![\phi]\!]_{I,\nu\oplus\langle x:v\rangle} = V$, et F sinon ;

(vii) $[\![\forall x\ \phi]\!]_{I,\nu}$ vaut V si pour toute valeur $v \in \mathbf{dom}$, $[\![\phi]\!]_{I,\nu\oplus\langle x:v\rangle} = V$, et F sinon.

NB : on dit qu'une valuation ν *satisfait* ϕ si $[\![\phi]\!]_{I,\nu} = V$. On dit qu'une formule est *satisfiable* s'il existe une valuation qui la satisfasse.

Définition 4.0.11 (Requête)**.**
Une requête est une expression de la forme

$$\{\ t_1, \ldots, t_n \mid \phi\ \}$$

où t_1, ..., t_n sont des termes, ϕ une formule et l'ensemble des variables apparaissant dans t_1, ..., t_n sont les éléments de $\mathrm{libre}(\phi)$.

La réponse à une telle requête est l'ensemble des $(\nu(t_1), \ldots, \nu(t_n))$ où ν est une valuation satisfaisant ϕ.

5 Exercices

Traduire dans le calcul relationnel les requêtes suivantes :

(i) Qui sont les réalisateurs ?

(ii) Quels sont les acteurs dont le prénom est Benedict ?

(iii) Quels sont les acteurs qui sont aussi des réalisateurs ?

(iv) Quels sont les acteurs qui ont joué au moins deux rôles dans un même film ?

(v) Quels sont les films dans lesquels le réalisateur est aussi un acteur ? [ambigu]

(vi) Quels sont les acteurs qui ont réalisé un film avant de jouer dans un autre ?

(vii) Quels sont les acteurs qui ont abandonné leur carrière d'acteurs pour se consacrer uniquement à la réalisation ? [ambigu]

6 À suivre

- Modèle relationnel : définition rigoureuse du MLD ;
- Calcul relationnel : modélisation des requêtes ;
- Peut-on traduire toute requête du calcul relationnel en SQL ?