SD314 : Exercices d'algèbre relationnelle

Author : Christophe Garion <christophe.garion@isae-supaero.fr>
Public : SUPAERO 3A SD/ARO
Date :

Solution

1. **Brune, blonde ou ambrée**

   Cet exercice est inspiré d'un examen écrit par S. Abiteboul pour le module de bases de données de l'ENS Cachan et de l'ENS ULM.

   On considère une base de données de buveurs de bière constituée de trois relations :

Frequente	(buveur, bar)
Sert	(bar, biere)
Aime	(buveur, biere)

   Frequente indique quels sont les bars fréquentés par un buveur de bière particulier, *Sert* indique quelles bières sont servies dans quels bars et *Aime* indique quelles bières sont appréciées par quel buveur. Des exemples de tables pour ces relations sont donnés sur les tableaux 1, 2 et 3.

buveur	**bar**
Alice	Le Filochard
Alice	L'Ancienne Belgique
Bob	La Tireuse
Bob	Le Bar de la Lune
Bob	Chez Tontongueu
Cecile	Le Filochard
Cecile	Chez Tontongueu
Cecile	La Tireuse
Martin	La Tireuse
Martin	Le Bar de la Lune
David	La Tireuse
David	Le Filochard
Emilie	La Tireuse

   TABLE 1 – Un exemple de relation *Frequente*

buveur	**biere**
Alice	Duvel
Bob	Duvel
Bob	Gueuze
Cecile	Gueuze
Martin	Jupiler
Martin	Leffe
David	Jupiler
David	Leffe
Emilie	Jupiler
Emilie	Leffe

   TABLE 2 – Un exemple de relation *Aime*

   Écrire en algèbre relationnelle les requêtes suivantes :

   (a) liste des bars qui servent une bière que Martin aime

   Solution :

   $$\pi_{bar}(\sigma_{buveur=Martin}(Sert \bowtie Aime))$$

   (b) liste des buveurs qui fréquentent au moins un bar qui sert une bière qu'ils aiment

bar	biere
Le Filochard	Biere Chouchen
Le Filochard	Leffe
Le Filochard	Jupiler
L'Ancienne Belgique	Leffe
L'Ancienne Belgique	Gueuze
La Tireuse	Jupiler
La Tireuse	Leffe
Chez Tontongueu	Duvel
Le Bar de la Lune	Gueuze

TABLE 3 – Un exemple de relation *Sert*

Solution :

$$\pi_{buveur}(Sert \bowtie Aime \bowtie Frequente)$$

L'utilisation simple de la jointure naturelle permettait de répondre à la question : $Aime \bowtie Buveur$ permet en effet de de créer une relation liant buveur, bière qu'il aime et bars fréquentés. La jointure de cette relation avec *Sert* permet ensuite de garder les bars servant les bières que le buveur aime.

2. **Une base de données de ventes**

Soit la base de données relationnelle de ventes de schéma suivant :

Client	(noClient, nom, noTel, adresse)
Article	(noArticle, description, prixUnitaire, qteEnStock)
Commande	(noCommande, dateCommande, noClient)
LigneCommande	(noCommande, noArticle, quantité)
Livraison	(noLivraison, dateLiv)
DetailLivraison	(noLivraison, noCommande, noArticle, qteLivree)

Les contraintes de clés étrangères sont les suivantes :

LigneCommande(noCommande)	référence Commande(noCommande)
LigneCommande(noArticle)	référence Article(noArticle)
DetailLivraison(noLivraison)	référence Livraison(noLivraison)
DetailLivraison(noCommande)	référence Commande(noLivraison)
DetailLivraison(noArticle)	référence Article(noArticle)

Écrire en algèbre relationnelle les requêtes suivantes :

(a) le numéro de commande et la date de commande des commandes du client 10 dont le numéro de commande est supérieur à 5.

Solution :

$\pi_{noCommande,dateCommande}(\sigma_{noClient=10 \ AND \ noCommande \geq 5} \ Commande)$

(b) les numéros des articles commandés au moins une fois par le client 10 après le 01/06/2000.

Solution :

$\pi_{noArticle}(\sigma_{dateCommande \geq 01/06/2000 \ AND \ noClient=10} Commande \ \bowtie LigneCommande)$

(c) les numéros de commande des commandes qui ont été passées à la même date que la commande 2.

Solution :

$\pi_{noCommande}(Commande \bowtie_{dateCommande} \pi_{dateCommande}((\sigma_{noCommande=2} Commande)))$

(d) Les numéros et noms des clients qui n'ont pas passé de commande au mois de mars de l'année 2000.

Solution :

$\pi_{noClient,nom} Client - \pi_{noClient,nom}(\sigma_{dateCommande \geq '2000-03-01' \ AND \ dateCommande \leq '2000-03-31'}(Client \bowtie Commande))$

(e) Les numéros des articles qui apparaissent dans toutes les commandes du client 10.

Solution :

$\pi_{noArticle}(NoCA) \ - \ \pi_{noArticle}(((\pi_{noArticle}(NoCA)) \times NoC) - NoCA)$

avec $NoC = \pi_{noCommande}(\sigma_{noClient=10}(Commande))$ et $NoCA = \pi_{noCommande,noArticle}(LigneCommande \bowtie NoC)$.

Je vous propose de détailler cette question. Le plus gros problème qui se pose est le fait que nous n'avons pas le droit d'utiliser des variables intermédiaires (on est en math, donc pas d'effet de bord).

Dans un premier temps, on peut trouver les numéros de commande du client 10 : $\pi_{noCommande}(\sigma_{noClient=10}(Commande))$. Appelons la relation obtenue *NoC*. On peut ensuite trouver pour chaque commande du client 10 les articles correspondants : $\pi_{noCommande,noArticle}(LigneCommande \bowtie NoC)$. Appelons la relation obtenue *NoCA*.

Prenons un exemple. Voici une instance de *NoC* :

noCommande
1
2
3

Voici une instance de *NoCA* :

noCommande	noArticle
1	100
1	101
2	100
2	101
2	105
3	100
3	101
3	106

Dans ce cas, on voit que les articles 100 et 101 apparaissent dans toutes les commandes du client 10. On devrait donc obtenir pour solution une table *Sol* de la forme suivante :

noArticle
100
101

C'est là où ça devient difficile. Quelle est la propriété vérifiée par *Sol* ? En fait, *Sol* est la table *maximale* (au sens du nombre de tuples contenus) telle que :

$$NoC \ \times \ Sol \subseteq NoCA$$

L'inclusion est ici à prendre au sens de l'inclusion des ensembles de tuples des relations. S'il n'y a pas d'articles apparaissant dans toutes les commandes, *Sol* ne possède pas de tuples. La condition de maximalité est nécessaire, car sinon *Sol* pourrait ne contenir que le premier tuple et ça marcherait.

Il nous reste donc à définir une opération permettant d'« inverser »[1] l'opération $\times$ pour trouver *Sol*. On va y aller doucement, en détaillant les étapes...

Tout d'abord, on remarque que $(\pi_{noArticle}(NoCA)) \times NoC$ nous donne une table reliant les numéros de commande à tous les articles commandés par le client 10 (dans notre exemple, on aurait ainsi un tuple $(1, 106)$ dans cette table). Donc $((\pi_{noArticle}(NoCA)) \times NoC) - NoCA$ nous donne une table dont les tuples (*noCommande*, *noArticle*) qui représentent le fait qu'un article n'apparaît pas dans une commande particulière. Donc $\pi_{noArticle}(((\pi_{noArticle}(NoCA)) \times NoC) - NoCA)$ nous donne les numéros d'articles qui n'apparaissent pas dans au moins une commande. La solution est donc donnée (ouf !) par :

$$\pi_{noArticle}(NoCA) \ - \ \pi_{noArticle}(((\pi_{noArticle}(NoCA)) \times NoC) - NoCA)$$

3. **Une base de données d'entreprise**
   Considérons une base de données relationnelle contenant des informations sur des employés. Elle possède plusieurs tables :

Employe	(NSS, Nom, Prenom, DateNaissance, Adresse, Salaire, NDep, Superieur)
Departement	(NDep, NomD, Directeur)
Projet	(NPro, NomP, Lieu, NDep)
Travaille	(NSS, NPro, Heures)

   Les contraintes de clés étrangères sont les suivantes :

Employe(Superieur)	référence Employe(NSS)
Employe(NDep)	référence Departement(NDep)
Departement(Directeur)	référence Employe(NSS)
Projet(NDep)	référence Departement(NDep)
Travaille(NPro)	référence Projet(NPro)
Travaille(NSS)	référence Employe(NSS)

   L'attribut `Superieur` dans la relation `Employe` contient le numéro de sécurité sociale du supérieur direct de l'employé. Chaque employé appartient à un département et travaille sur un ou plusieurs projets. Chaque projet est rattaché à un département qui peut être *différent* de celui des employés travaillant sur ce projet.

   Écrire en algèbre relationnelle les requêtes suivantes :
   (a) date de naissance et adresse de Alice Dupont
   Solution :
   $\pi_{DateNaissance,\ Adresse}(\sigma_{Prenom='Alice'\ AND\ Nom='Dupont'}(Employe))$

   (b) nom et adresse des employés qui travaillent au département de recherche
   Solution :
   $\pi_{Nom,\ Adresse}(\sigma_{NomD='Recherche'}(Employe \bowtie Departement))$

   (c) nom des employés qui travaillent plus de 10 heures sur un projet à Genève
   Solution :
   $\pi_{Nom}(\sigma_{Heures>10\ AND\ Lieu='Genève'}((Projet \bowtie Travaille) \bowtie_{NSS\ =\ Employe.NSS} Employe))$

   (d) nom et prénom des employés qui ne travaillent sur aucun projet
   Solution :
   $\pi_{Nom,\ Prenom}(\pi_{Nom,\ Prenom,\ NSS}(Employe) - \pi_{Nom,\ Prenom,\ NSS}(Travaille \bowtie Employe))$

License CC BY-NC-SA 3.0

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported license (CC BY-NC-SA 3.0)

You are free to Share (copy, distribute and transmite) and to Remix (adapt) this work under the following conditions:

Attribution – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Noncommercial – You may not use this work for commercial purposes.

Share Alike – If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

See http://creativecommons.org/licenses/by-nc-sa/3.0/ for more details.