

Relations en non première forme normale pour représenter hiérarchiquement des données structurées

Serge Abiteboul

Nicole Bidoit

PODS 1984

Les bases de données

Une base de données permet :

- stockage d'une masse d'informations.
- optimisation de la gestion des données : ajout, interrogation ...
- gestion des 12 problèmes BD : reprise sur panne, contrôle de concurrence, contraintes d'intégrité, confidentialité, indépendance des niveaux. ...

Plusieurs modèles de données pour représenter le niveau logique

- Modèle hiérarchique - 1960
- Modèle réseau - fin 60's
- Modèle relationnel - 1970

Plan

- 1 Introduction
 - Les bases de données
 - Le modèle relationnel
- 2 Le problème
- 3 Mise en œuvre
 - Le modèle de données Verso
 - Opérations sur le modèle Verso
- 4 Travaux reliés
- 5 Conclusion

Exemple - Incomplet

Énoncé de l'exemple

On veut représenter par un modèle de données l'application associant à chaque **cours**, l'**enseignant** qui l'assure et l'ensemble des **étudiants** qui y sont inscrits.

En relationnel, on aura besoins de deux relations

- $\mathfrak{R}_1(\text{cours}, \text{enseignant})$, associant à chaque cours son enseignant.
- $\mathfrak{R}_2(\text{cours}, \text{etudiant})$, associant à chaque cours ses étudiants.

Plan

- 1 Introduction
 - Les bases de données
 - Le modèle relationnel
- 2 Le problème
- 3 Mise en œuvre
 - Le modèle de données Verso
 - Opérations sur le modèle Verso
- 4 Travaux reliés
- 5 Conclusion

Le modèle relationnel

Fondement

C'est un modèle introduit en 1970 par E. F. Codd, basé sur un concept mathématique (algèbre relationnelle)

L'algèbre relationnelle

- la relation (la table) est la notion centrale.
- une relation \mathfrak{R} est spécifiée par :
 - un ensemble d'attributs $A_1 \dots A_n$ qui constituent son schéma : $\mathfrak{R}(A_1, A_2, \dots, A_n)$ s
 - une instance $\exists \subseteq \{dom(A_1) \times dom(A_2) \times \dots \times dom(A_n)\}$
- des opérations ensemblistes et de restriction permettent l'extractions des données.

Optimisation

contraintes sur les relations et les attributs.

Optimisation de la gestion des données

Normalisation

Elle est liée au modèle relationnel et permet d'avoir des relations dites en **forme normale**. Cela permet de :

- supprimer les redondances et faciliter la mise à jour.
- faciliter le stockage des données.
- améliorer la modélisation et la lisibilité.

1NF - Définition

- Une relation est en première forme normale si tout ses attributs n'ont que des valeurs atomiques.
- Une valeur est atomique si elle ne peut pas être divisée en sous-valeurs

Inconvénient

Temps d'accès plus long si les requêtes sont trop complexes.

Énoncé du problème

- Introduire un modèle de données :
 - plus souple.
 - basé sur des relations.
 - plus optimal pour la gestion de certaines applications.
- Étendre l'algèbre relationnelle à ce nouveau modèle.

- Laisser à l'utilisateur le choix d'imbriquer des relations afin de gagner en optimisation.
- Optimisation et gain de temps car certaines requêtes dans le modèle Verso nécessiteront des simples sélection, tandis qu'elle nécessiteront une jointure (plus couteuse) dans le modèle relationnel.

- 1 Introduction
 - Les bases de données
 - Le modèle relationnel
- 2 Le problème
- 3 Mise en œuvre
 - Le modèle de données Verso
 - Opérations sur le modèle Verso
- 4 Travaux reliés
- 5 Conclusion

Format

Chaîne d'attributs

Une chaîne de caractère finie d'attributs est de la forme $X_1 X_2 \dots X_n$ où chaque X_i est un attribut.

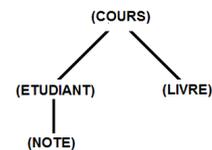
Format

Un format est défini récursivement par :

- Soit X une chaîne d'attributs sans répétition d'attributs. Alors X est un **format plat** sur l'ensemble X des attributs.
- Soit X une chaîne d'attributs sans répétition d'attributs, et f_1, \dots, f_n des formats sur Y_1, \dots, Y_n resp. tels que les ensembles X, Y_1, \dots, Y_n soient deux à deux disjoints. Alors la chaîne $X(f_1) \star \dots (f_n)$ est un **format** sur l'ensemble $X Y_1 \dots Y_n$.

Format - exemple et représentation

- $f = \text{cours etud note}$ est un format plat sur $\{\text{cours, etud, note}\}$
- $g = \text{cours(etud(note)\star)(livre)\star}$ est un format sur $\{\text{cours, etud, note, livre}\}$
- ci-dessous, la représentation du format g sous forme arborescente :



Instance / schéma (1/2)

Ensemble des instances

Soit f un format. L'ensemble des instances (Verso) $inst(f)$ sur f , est défini par :

- Si $f \equiv X_1 \dots X_n$ (est un format plat) :
 - $inst(f) = \text{ensembles des parties de } \{dom(X_1) \times dom(X_2) \times \dots \times dom(X_n)\}$
- Si $f \equiv X_1 \dots X_n(f_1) \star \dots (f_n)$:
 - 1 \mathfrak{S} est un sous-ensemble fini de $\{dom(X_1) \times dom(X_2) \times \dots \times dom(X_n)\} \times inst(f_1) \times \dots \times inst(f_n)$
 - 2 si $\langle u, l_1, \dots, l_n \rangle$ et $\langle u', l_1, \dots, l_n \rangle$ sont dans \mathfrak{S} alors $u = u' \Rightarrow \langle u, l_1, \dots, l_n \rangle = \langle u', l_1, \dots, l_n \rangle$

Instance / schéma (2/2)

Schéma

Un schéma de base de données Verso S est un ensemble fini de formats.

Instance

Une instance de base de données Verso s est une application de S dans $\bigcup_{f \in S} inst(f)$ telle que : $s(f)$ est une instance sur le format f pour tout f dans S .

Les opérations Verso (1/2)

La jointure

Les opérations Verso (2/2)

La sélection

Travaux reliés

- *Verso : A Relational Back End Data Base Machine* :
 - F. Bancilhon & al.
 - Workshop on Database Machines, San Diego, 1982.

En bref...

Résumé

Le modèle Verso est un Modèle BD basé sur des relations en non-1NF. De ce fait, un simple langage algébrique peut-être défini pour ce modèle.

Perspectives

(?) A voir.

Plan

- 1 Introduction
 - Les bases de données
 - Le modèle relationnel
- 2 Le problème
- 3 Mise en œuvre
 - Le modèle de données Verso
 - Opérations sur le modèle Verso
- 4 Travaux reliés
- 5 Conclusion

Plan

- 1 Introduction
 - Les bases de données
 - Le modèle relationnel
- 2 Le problème
- 3 Mise en œuvre
 - Le modèle de données Verso
 - Opérations sur le modèle Verso
- 4 Travaux reliés
- 5 Conclusion