Modélisation de données - 3IF

Victor Lezaud

7 novembre 2017

Sommaire

1 Modèle Relationnel

1.1 Théorie

Dans cette partie on cherche à modéliser les données d'employés avec leurs prénoms, noms, âges dans différents départements ainsi que leurs activités

Le but du modèle relationnel est de stocker dans des tableaux en deux dimensions des données décrivant un ensemble d'attributs nommé univers. Dans notre exemple l'univers est $U = \{nss, nom, prenom, age, dep, adresse, activite\}$

Définition de symbole de relation : Un symbole de relation $SdeR$ représente la première ligne d'un tableau. On définit les fonctions suivantes

- $schema(SdeR)$ est l'ensemble des attributs de $SdeR$
- $type(SdeR)$ est le nombre d'attributs de $SdeR$
- $attr_i(SdeR)$ est le i$^{\text{ième}}$ attribut de $SdeR$

Définition de tuple : Un tuple correspond à une ligne dans un tableau à double entrée. Un tuple sur $SdeR$ est une liste de valeurs pour tous les attributs de $SdeR$. Par exemple (1, "INFORMATIQUE", "Bâtiment Blaise Pascal") est un tuple sur $DEPT$.

Définition de relation : Une relation sur un symbole de $SdeR$ est un ensemble de tuple sur $SdeR$. ATTENTION : ne pas confondre relation est symbole de relation ! ! La relation est l'ensemble des valeurs qui sont dans le tableau tandis que le symbole de de relation est le squelette

Définition de domaine et domaine actif : Le domaine d'un attribut est l'ensemble des valeurs que peut prendre un attribut. Le domaine actif d'un attribut A appartenant à $schema(SdeR)$ dans la relation r noté $ADOM(A, r)$, est l'ensemble des valeurs prises par A dans r. On a $ADOM(A, r) = \pi_A(r)$.

Définition de base de données : Un symbole de base de données est un ensemble de symbole de relation et une base de données est un ensemble de relation.

Hypothèse de nommage : Universal Relation Schema Assumption (URSA) : si un attribut apparait dans plusieurs schémas de relation, alors cet attribut représente les même données.

1.2 Exemple

Soit le symbole de bd $BD = \{PERSONNE, DEPT, TRAVAILLENT\}$ avec $schema(PERSONNE) = \{nss, nom, prenom, age\}$, $schema(DEPT) =$

$\{dep, adresse\}$ et $schema(TRAVAILLENT) = \{nss, dep, activite\}$.

La base de données bd sur BD avec les relations Personnes sur $PERSONNE$, $Departements$ sur DEP et $Travaillent$ sur $TRAVAILLENT$ avec quelques tuples.

Personnes	nss	nom	prenom	age
	12	Aymard	Serge	45
	45	Fenouil	Solange	35

Departements	dep	adresse
	Math	Carnot
	Info	Cézeaux

Travaillent	nss	dep	activite
	12	Math	Prof
	45	Math	MdC
	45	Info	MdC

2 Langages de requêtes

2.1 Les différents langages

Il existe deux approches pour les requêtes sur les bases de données relationnelles

Approche algébrique : Algèbre relationnelle Effectue des transformation de relation (opération unaire et binaire), impose un ordre d'exécution de la requête

Approche logique : Calcul relationnel Sélectionne les tuples vérifiant des formules logique à partir de l'ensemble des possibles (produit cartésien des domaines des attributs)

Langage SGBD : SQL Mélange des deux approches, sucre syntaxique

2.2 La projection

La projection est une coupe verticale dans la relation. Soient r une relation sur R et $Y \subseteq schema(R)$. La projection de r sur Y, notée $\pi_Y(r)$, est définie par :

$$\pi_Y(r) = \{t[Y] | t \in r\}$$

2.3 La sélection

La sélection ne garde que les tuples qui vérifient une formule de sélection, qui est formée par une combinaison d'opérateurs (et, ou, non) logiques et de formule simple (comparaison).
Soient r une relation sur R et F une formule de sélection sur R. La sélection des tuples de r par rapport à F, notée $\sigma_F(r)$, est définie par :

$$\sigma_F(r) = \{t \in r | t \models F\}$$

2.4 Opérations ensemblistes

Union : $r1 \cup r2 = \{t | t \in r1 \ \vee \ t \in r2\}$

Différence : $r1 - r2 = \{t | t \in r1 \ \wedge \ t \notin r2\}$

Intersection : $r1 \cap r2 = \{t | t \in r1 \ \wedge \ t \in r2\}$

2.5 La jointure

Soient r_1 et r_2 deux relations sur R_1 et R_2 respectivement

La jointure naturelle de r_1 et r_2, notée $r_1 \bowtie r_2$, est une relation sur un symbole de relation R, avec $schema(R) = schema(R_1) \cup schema(R_2)$, définie par :

$$r_1 \bowtie r_2 = \{t \mid \exists t_1 \in r_1 \ \exists t_2 \in r_2 \ tq \ t[schema(R_1)] = t_1 \ et \ t[schema(R_2)] = t_2\}$$

2.6 Le renommage

Le renommage n'a d'intérêt qu'en algèbre relationnelle pour réaliser les jointures souhaitées. Soit r une relation sur R avec $A \in schema(R)$ et $B \notin schema(R)$.
Le renommage de A en B dans r, noté $\rho_{A \to B}$, est une relation sur S avec $schema(S) = (schema(R) - \{A\}) \cup \{B\}$.

2.7 La division

But : La division permet de sélectionner les tuples associés à un autre ensemble de tuples, par exemple elle permet de répondre à la question : "Quels sont les étudiants qui sont inscrits dans tous les départements".

Définition de la division : Soient r une relation sur R avec $schema(R) = \{X, Y\}$ et s une relation sur S avec $schema(S) = \{Y\}$. La division de r par s, notée $r \div s$, est une relation sur un symbole de relation R_1, avec $schema(R_1) = \{X\}$, définie par :

$$r \div s = \{t[X] | t \in r \ et \ s \subseteq \pi_Y(\sigma_{F(t)}(r))\}$$

avec $X = \{A_1, \cdots, A_q\}$ et $F(t) = (A_1 = t[A_1]) \wedge \cdots \wedge (A_q = t[A_q])$

Équivalence avec les autres opérateurs :

$$r \div s = \pi_X(r) - \pi_X((\pi_X(r) \times s) - r)$$

Division en calcul relationnel

$$r \div s = Ans(Q, \{r, s\} avec Q = \{< x : X > | \forall y : S(R(x, y) \vee \neg S(y))\}$$

2.8 Variables positives et négatives

Une variable est positive si elle a un nombre fini de valeur possible. Une variable est négative si elle peut prendre un nombre infinie de valeur.x

Variable positive

Une variable x est positive dans les formules atomiques suivantes :

— $R(y_1, \cdots, x, \cdots, y_k)$

— $x = cte$

Les opérateurs rendent x positives si :

— $\neg F$, si F est atomique et x négative dans F

— $F_1 \wedge F_2$, si x est positive dans F_1 OU F_2

— $F_1 \vee F_2$, si x est positive dans F_1 ET F_2

— $\exists y : A(F)$, si $x \neq y$ et x positive dans F

Variable négative

Une variable x est négative dans les formules atomiques suivantes :

— $x = y$ avec y une variable négative

Les opérateurs rendent x négative si :

— $\neg F$, si F est atomique et x positive dans F

— $F_1 \wedge F_2$, si x est négative dans F_1 ET F_2

— $F_1 \vee F_2$, si x est négative dans F_1 OU F_2

— $\forall y : A(F)$, si $x \neq y$ et x négative dans F

— si x n'apparaît pas dans F

2.9 Formule de calcul autorisée

Une formule de calcul F est autorisée si :

— Chaque variable libre de F est positive dans F

— Pour chaque sous-formule $\exists x : A(G)$ de F, x est positive dans G

— Pour chaque sous-formule $\forall x : A(G)$ de F, x est négative dans G

3 Les dépendances

Dans cette partie nous allons voir comment contraindre la base de données afin de n'avoir que des données valides.

3.1 Dépendances Fonctionnelles

Principe des DF : Les DF permettent de forcer une base de données à suivre une logique d'implication et d'éviter par exemple que deux personnes aient un même numéro de sécurité sociale ou qu'une même personne ait deux moyennes pour un même cours. Les DF sont liés au principe de clés.

Définition d'une DF : Soit R un symbole de relation. Une DF sur R est une déclaration de la forme : $R : X \to Y$, où $X, Y \subseteq schema(R)$

Satisfaction d'une DF : Soit r une relation sur R. La satisfaction de la DF $R : X \to Y$ par r est notée $r \models X \to Y$ et vérifie :

$$r \models X \to Y \Leftrightarrow \forall t_1, t_2 \in r,\ si\ t_1[X] = t_2[X]\ alors\ t_1[Y] = t_2[Y]$$

Définition d'une clé : Une superclé est un ens. d'attributs $X \subseteq schema(R)$ qui vérifie la DF $X \to schema(R)$. Une clé (ou clé minimale) est une superclé vérifiant $\nexists Y \subset X$, Y superclé de R.

3.2 Dépendances d'Inclusion

Principe des DI : Les DI permettent de forcer les attributs d'une relation à avoir des valeurs présentent dans une autre relation de la base de données. Cela permet d'éviter que l'on donne une note à un élève pour un cours totalement inconnu de la base de données.

Définition des DI : Soit R un symbole de base de données avec R_1 et R_2 deux symboles de relations de R. Une DI sur R est de la forme $R_1[X] \subseteq R_2[Y]$, avec X et Y des séquences d'attributs appartenant respectivement à $schema(R_1)$ et $schema(R_2)$.

Satisfaction des DI : Soient d une base de données sur R et $r_1, r_2 \in d$ définies respectivement sur $R_1, R_2 \in R$. La satisfaction de la DI $R_1[X] \subseteq R_2[Y]$ par d est notée $d \models R_1[X] \subseteq R_2[Y]$ et vérifie :

$$d \models R_1[X] \subseteq R_2[Y] \Leftrightarrow \forall t_1 \in r_1, \exists t_2 \in r_2\ tq\ t_1[X] = t_2[Y]$$

Définition d'une clé étrangère : Une clé étrangère est un ensemble d'attributs X de R_1 intervenant dans une DI $R_1[X] \subseteq R_2[Y]$ avec Y une superclé de R_2.

3.3 Projection de F sur un sous-ensemble d'attributs

Soient U l'univers, F un ensemble de DF sur U et $S \subset U$. La projection de F sur S, notée $F[S]$, est définie par :

$$F[S] = \{X \to Y \mid F \models X \to Y,\ X \cup Y \subseteq S\}$$

3.4 Fermeture et fermé

Soit F un ensemble de DF sur R et $X \subseteq schema(R)$

Définition de fermeture : La fermeture de X par rapport à F, notée X_F^+, est l'ensemble des attributs fixés quand X est fixé soit :

$$X_F^+ = \{A \in schema(R) | F \models X \to A\}$$

Définition de fermé : X est un fermé de F si $X = X_F^+$, autrement-dit si il n'implique que lui-même.

CL(F) et IRR(F) : On note $CL(F)$ l'ensemble des fermés de F et $IRR(F)$ ses éléments irréductibles par intersection (que l'on ne peut pas obtenir par intersection des autres).

$$CL(F) = \{X | X = X_F^+\}$$

$$IRR(F) = \{X \in CL(F) | \forall Y, Z \in CL(F)\ (X = Y \cap Z) \Rightarrow (X = Y\ ou\ X = Z)\}$$

3.5 Relation d'Armstrong

Définition de la relation d'Armstrong : Soit r une relation sur R, F un ensemble de DF sur R. r est une relation d'Armstrong par rapport à F si elle vérifie l'équivalence suivante :

$$(r \models X \rightarrow Y) \Leftrightarrow (F \vdash X \rightarrow Y)$$

Construction d'une relation d'Armstrong :

1. calculer $CL(F)$
2. Créer la relation r avec un tuple ne contenant que des 0
3. (étape i) pour chaque élément X de $CL(F)$, ajouter un tuple t_i à r tq :
   — $t_i[A] = 0$ si $A \in X$
   — $t_i[B] = i$ si $B \notin X$

3.6 Couverture d'ensemble de DF

Définition de la couverture : Une couverture F (ou base) d'un ensemble K de DF est un ensemble de DF équivalent à K, noté $K^+ = F^+$.

Caractéristique d'unes d'une couverture : Une couverture F de K est dite :

— non redondante si $F \subseteq K$, $\nexists G \subset F$ tel que $G^+ = F^+$
— minimale si $\nexists G\ tel\ que\ G^+ = F^+\ et\ |G| < |F|$

4 Problème d'implication

4.1 Introduction

Deux ensembles de DF peuvent être écrit différemment mais parfaitement équivalent. On va donc se demander dans ce chapitre si un ensemble F de DF sur R implique une DF $X \rightarrow Y$ ou non. Autrement dit, peut on déduire ou prouver $X \rightarrow Y$ à partir de F. Pour cela il existe trois approches totalement équivalentes : la théorie des modèles et la théorie de la preuve.

4.2 Théorie des modèles

Concept : La théorie des modèles travaille au niveau des relations, elle consiste à observer toutes les relations sur R qui vérifient F et tester si elles vérifient $X \rightarrow Y$.

Exprimé mathématiquement : Soient F un ensemble de DF sur R et $X \to Y$ une DF sur R alors on a l'équivalence suivante :

$$F \models X \to Y \Leftrightarrow (\forall r \; sur \; R, \; r \models F \Rightarrow r \models X \to Y)$$

4.3 Théorie de la preuve

Concept : La théorie de la preuve ne travaille qu'avec l'ensemble de Dépendances Fonctionnelles et le symbole de relation, elle consiste à créer une preuve à partir des différentes DF et des règles d'inférence d'Armstrong

Règles d'inférence d'Armstrong
Réflexivité : si $Y \subseteq X \subseteq schema(R)$, alors $F \vdash X \to Y$
Augmentat° : si $F \vdash X \to Y$ et $W \subseteq schema(R)$, alors $F \vdash XW \to YW$
Transitivité : si $F \vdash X \to Y$ et $F \vdash Y \to Z$, alors $F \vdash X \to Z$

Bonus : propriété des fermetures

Cette propriété est peu utilisé car on a souvent besoin de résoudre les problèmes avant de trouver les fermetures de F, mais elle reste bonne à savoir car très simple :

$$(F \models X \to Y) \Leftrightarrow (Y \subseteq X_F^+)$$

5 Conception de Base de données

5.1 Les Formes Normales

Principe des Formes Normales : Les FN permettent de spécifier formellement la notion intuitive de bon schéma. Plusieurs FN existe : 1FN (moins restrictive), 2FN, 3FN, FN de Boyce-Codd (FNBC) (plus restrictive).

Définition de la Forme Normale de Boyce-Codd : Un symbole de relation R est en FNBC par rapport à un ensemble F de DF si et seulement si pour toutes les dépendances de F, leurs parties gauches sont des superclés. Le but est de ne représenter l'information qu'une seule fois.

5.2 Algorithme de synthèse

But : L'algorithme de synthèse permet de créer un schéma de base de données à partir des dépendances fonctionnels sur l'univers. Les schémas ainsi créés sont en 3FN.

Algorithme :

1. Déterminer une couverture minimale G de F
2. Réduire les parties gauches et droites des DF de G
3. Créer un symbole de relation pour chaque DF de G avec les attributs impliqués dans cette DF

5.3 Algorithme de décomposition

But : L'algorithme de décomposition permet de créer un schéma de base de données à partir des dépendances fonctionnels sur l'univers. Les schémas ainsi créés sont en FNBC.

Algorithme : L'algorithme de décomposition est un algorithme récursif qui prend en entrée un ensemble R d'attributs et un ensemble F de DF. Si on est en FNBC alors l'algorithme s'arrête. Sinon on prend une DF sans superclé dans la partie gauche, on crée un symbole de relation avec les attributs de cette DF. Puis on appelle l'algorithme avec ce nouveau symbole de relation et la projection de F sur ce symbole de relation. On appelle aussi l'algorithme avec le reste du symbole de relation et la projection de F sur ce reste.

Algorithme Décompose(R, F)

SI R est en FNBC par rapport à F ALORS
——retourne R
SINON
——Soit $X \to Y$ une DF non triviale de F tel que $X_F^+ \neq R$
——Décompose(XY,$F[XY]$)
——Décompose($R - (Y - X)$, $F[R - (Y - X)]$)