Introduction aux systèmes d'information et aux bases de Données, Partie 2 : Raffiner des schémas de bases de données

L2 Informatique

S. Cerrito

premier sémestre 2014-2015

Si on suit la méthode EA, normalement on arrive à une BD ayant des bonnes propriétés.

Mail peut arriver qu'on ait besoin de corriger une base contenant des tables problématiques, qu'il faudra décomposer, selon certains critéres.

0.1 Décompositions d'un schéma : propriétés SPI et SPD

Voici un mauvais schéma de table :

Relation FOURNISSEUR

Nom	Adr	$Produit$	$Prix$
n1	a1	i1	p1
n1	a1	i2	p2
n2	a2	i2	p3
n2	a2	i3	p4

$F = Nom \rightarrow Adr,\ Nom\ Produit \rightarrow Prix$

La clé est $Nom, Produit$, mais Adr ne dépend que de Nom, ce qui porte à répéter inutilement, l'adresse d'un producteur pour chaque produit qu'il fournit. Risque d'incohérences de mise à jour.

Il faudrais le remplacer par un autre schéma. Comment ?

Une solution possible : décomposer le schéma de relation en 2 ou plusieurs sous-schémas. Par ex. décomposer :

$$\text{FOURNISSEUR}(Nom,\ Adr,\ Produit,\ Prix)$$

en :

$$\text{NA}(Nom,\ Adr) \text{ et } \text{NIP}(Nom,\ Produit,\ Prix)$$

Contenu de la table NA : $\pi_{Nom,Adr}(\text{FOURNISSEUR})$.
Contenu de la table NIP : $\pi_{Nom,Produit,Prix}(\text{FOURNISSEUR})$.

Ceci est un exemple de la façon générale de décomposer une table de schéma S en n tables de schémas $S_1, \cdots, S_n$ où $(S_1 \cup \cdots \cup S_n) = S$:
on projette le contenu de la table decomposée sur les sous-schémas.

- **Toute décomposition possible est OK** ?
- Un des critères qu'une bonne décomposition doit respecter : il faut pouvoir reconstruire, par jointure, la relation de départ.
- Dans notre exemple :
  $\pi_{Nom,Adr}(FOURNISSEUR) \bowtie \pi_{Nom,Produit,Prix}(FOURNISSEUR) = FOURNISSEUR$? OUI.
- Toujours vrai, peu importe la décomposition ? NON. Contre-ex : Table R :

A	B	C
a1	b1	c1
a2	b1	c2

  Décomposition en R1(A,B) et R2(B,C) (contenus=projections de R).
  $\langle a1, b1, c2\rangle \in (\pi_{A,B}(R) \bowtie \pi_{B,C}(R))$ mais $\langle a1, b1, c2\rangle \notin$ R !
  Cette décomposition fait perdre l'information NEGATIVE :
  "$\langle a1, b1, c2\rangle$ **n'est pas** une ligne de R".

- Soit F un ensemble de DF. Une décomposition d'un schéma S d'une table R en $S_1, \cdots, S_n$ (où $(S_1 \cup \cdots \cup S_n) = S$) est dite sans perte d'information (SPI) par rapport à F ssi, qque soit le contenu de la table de nom R :

$$(\pi_{S_1}(R) \bowtie \cdots \bowtie \pi_{S_n}(R)) = R$$

- *N.B.* :

$$R \subseteq (\pi_{S_1}(R) \bowtie \cdots \bowtie \pi_{S_n}(R))$$

  est toujours vrai, la $\subseteq$ réciproque non (contre-exemple précédant).

Rôle de F ??

$\Rightarrow$

Pour le schéma de table de l'example précédant, on n'aurait pas pu avoir perte d'information négative si $B \to C$ ou $B \to A$:
on obtient $\langle a1, b1, c2\rangle \in (\pi_{A,B}(R) \bowtie \pi_{B,C}(R))$ mais $\langle a1, b1, c2\rangle \notin R$ car R ne satisfait pas $B \to C$ et ne satisfait pas $B \to A$

Soit $S = \{A, B, C\}$ un schéma de table et soit F un ensemble de DF sur S.
Si F implique $B \to C$ ou $B \to A$, alors la déc. de S en $\{A, B\}$ et $\{B, C\}$ est forcement SPI par rapport à F.

Remarque :
$B \to A = (\{A, B\} \cap \{B, C\} \to (\{A, B\} \setminus \{B, C\}$
$B \to C = (\{A, B\} \cap \{B, C\} \to (\{B, C\} \setminus \{A, B\}$

Théorème : Une déc. de S en red 2 sous-schémas S_1 et S_2 est SPI par rapport à un ensemble de dépendances fonctionnelles F ssi F implique

$$(S_1 \cap S_2) \rightarrow (S_1 \setminus S_2)$$

ou

$$(S_1 \cap S_2) \rightarrow (S_2 \setminus S_1)$$

NB : ce théorème donne un algo pour tester si une décomposition en 2 est SPI par rapport à F.

Et si on veut décomposer un schéma de table S en n schémas de tables où $n > 2$ et tester si cette déc. est SPI ? Algorithme dit chase.

Algorithme Chase

Conventions :

- Nous supposons que les attributs de S sont $A_1, ..., A_k$, où $k \geq 2$;
- Nous associons à chaque attribut A_i une et une seule *constante* a_i
- Les expressions de la forme $x^i{}_j$ sont des *variables*

Entrées : Le schéma S, les schémas de sa décomposition : $S_1, \cdots, S_n$, l'ensemble de dépendances fonctionnelles F. Sans perte de généralité, on peut supposer que toute dépendance de F est réduite à droite (il y a un seul attribut à droite de $\rightarrow$)
Sortie : "OUI", si la décomposition est SPI par rapport à F, "NON" sinon.

1. *Initialisation.*

   a) Construire une table dont les k colonnes sont $A_1, ..., A_k$ et ayant n lignes. Chaque ligne j est étiquetée par S_j :

	A_1	$\cdots$	A_i	$\cdots$	A_n
S_1					
$\vdots$					
S_j					
$\vdots$					
S_n					

   b) Dans chaque case de coordonnées i, j (i est la colonne, j est la ligne), placer la constante a_i si l'attribut $A_i \in S_j$, placer la variable $x^i{}_j$ sinon.

2. **Jusqu'à** *ou bien on obtient au moins une ligne ne contenant aucune variable, ou bien la table reste stable*, **faire** :

   pour chaque dépendance $X \to A_i$ de F,

pour chaque couple de lignes,
si ces deux lignes ont les mêmes valeurs pour X mais des valeurs différents pour A_i, alors modifier ces lignes de façon qu'elles aient la même valeur pour A_i aussi.
Pour effectuer cette modification, si un ligne contient une variable pour la colonne A_i et l'autre la constante a_i, remplacer la variable par la constante, tandis que si les deux lignes contiennent deux variables pour la colonne A_i, remplacer une des 2 variables (peu importe laquelle) par l'autre.

3. Si table finale contient au moins une ligne ne contenant aucune variable, la sortie est "OUI", sinon c'est "NON".

Exemple

Soit $S = ABCDE$ et soit F l'ensemble de dépendances :

$$\{A \to C, B \to C, C \to D, DE \to C, CE \to A\}$$

Appliquer l'algorithme *chase* pour déterminer si la décomposition AD, AB, BE, CDE, AE est SPI par rapport à F.

Critère 1 qu'une décomposition doit respecter : être SPI par rapport aux DF.

- SPI : seul critère à souhaiter pour une déc. d ?
- **Exemple**. V : ville, R : rue, C : Code Postal. Schéma : INFOS(R,C,V).
  $F = \{VR \to C,\ C \to V\}$.
  Déc. d en INFO1(RC) et INFO2(CV) :

	INFOS	
R	C	V
r1	c1	v1
r1	c2	v2
r3	c3	v3

INFO1	
R	C
r1	c1
r1	c2
r3	c3

INFO2	
C	V
c1	v1
c2	v2
c3	v1

  d est SPI car F implique $((\text{RC}) \cap (\text{CV})) \to (\text{CV} \setminus \text{RC})$.

  Mais que dévient $VR \to C$? Pas applicable.

  Insertion de $\langle r1, c3\rangle$ dans INFO1 : OK par rapport aux dép de INFO1, mais :
  $\langle r1, c3, v1\rangle$ **et** $\langle r1, c1, v1\rangle \in \text{INFO1} \bowtie \text{INFO2}$. Violation de $VR \to C$!

La décomposition l'exemple précédent ne fait pas perdre d'information mais elle fait perdre des dépendances fonctionnelles.

Pour savoir si une insertion préserve la contrainte $VR \to C$ il faudrait réfaire la jointure INFO1 $\bowtie$ INFO2 !

On aimerait travailler sans besoin de faire des $\bowtie$ chaque fois qu'on insère.

Formalisons : $\Rightarrow$

Déf. Une dép. fonct. $X \to Y$ est *applicable* sur un schéma de relation S_i ssi $X \cup Y \subseteq S_i$.

Déf. La *fermeture* d'un ensemble de DF F, noté F^+, est l'ensemble de *toutes* les dép. impliquées par F.

Déf. Deux ensembles de DF F_1 et F_2 sont *équivalents* ssi $F_1^+ = F_2^+$.

Déf. La *projection* d'un ensemble de dép. F su'un ensemble d'attributs X est l'ensemble d'éléments de F^+ (pas seulement de F !) comportant seulement des attributs de X. Notation : F_X.

Déf. Une décomposition d de S en $S_1 \cdots S_n$ est sans perte de dépendances (SPD) par rapport à un ensemble de dép. F ssi

$$(F_{S_1} \cup \cdots \cup F_{S_n})^+ = F^+$$

NB : $\subseteq$ est banal. C'est l'inclusion reciproque qui compte.
(Le pb. crucial est : risque-t-on de perdre des dépendances impliquées par F, en projettant sur les sous-schémas ?)

Exemple de calcul de F_{S_i}. Soit $S = \{A, B, C\}$, $S_1 = \{A, C\}$

$F = \{A \to B, B \to A, B \to C, C \to B\}$.

$F^+ = \{A \to A,\ A \to B,\ A \to C,$
$B \to A,\ B \to B,\ B \to C,$
$C \to A,\ C \to B,\ C \to C, ...\}$

F_{S_1} contient $A \to C$ et $C \to A$ (même si $C \to A \notin F$!) et toutes les dépendances impliquées par ces deux la.

Attention : Oublier que $A \to C$ et $C \to A$ sont dans F_{S_1} est un exemple typique d'erreur !

Comment calculer de façon mécanique si f est impliquée par un ensemble de dép. fonct. F ?

- **Déf.** : Soit Z l'ensemble des attributs dans F, $X \subseteq Z$.
  La *fermeture* de X par rapport à F ($X^+{}_F$) est :

$$max\{Y \subseteq Z \mid F \; implique \; X \rightarrow Y\}$$

- **Théorème** : F implique $X \rightarrow Y$ ssi $Y \subseteq X^+{}_F$.
- Donc, pour savoir si F implique $X \rightarrow Y$ il suffit de savoir calculer $X^+{}_F$!.

ALGO DE CALCUL DE $X^+{}_F$:

Entrée : $F,\ X,\ Att$, où Att est l'ensemble de tous les attributs dans F
Sortie : $X^+{}_F$ comme valeur de la variable res.

res:=X;
répéter :
pour chaque $Y \to Z \in F$, **si** $Y \subseteq res$ **alors** $res := res \cup Z$
jusqu'à quand res n'a pas changé ou $res = Att$;
renvoyer res

Exemple de calcul de $AB^+{}_F$ pour $F = \{A \to C, BC \to D, ED \to E\} \rightsquigarrow$ tableau.