2. Décomposition d'un schéma relationnel

i. Introduction

on dit qu'un schéma R est décomposé en sous schémas R1, R2, ... Rk lorsque la réunion des attributs des Ri est égal à l'ensemble des attributs de R.
$U = X_1 \cup X_2 \ldots \cup X_k$

Exemple 1 :

ecritPar(numéro_livre, titre_livre,nb_page, auteur_nom, auteur_adresse, éditeur, éditeur_adresse)
peut être décomposée en :

R1(numéro_livre, titre_livre,nb_page, auteur_nom)
R2(auteur_nom, auteur_adresse)
R3(éditeur, éditeur_adresse)

Car la réunion de tous les attributs des Ri redonne bien les attributs de écritPar.

ii. Décomposition sans perte d'information

On dit que la décomposition et sans perte d'information, notée SPI, si et seulement si pour toute instance r de R :
Si ri = proj(r) / Xi alors r = r1►◄r2►◄.. ►◄rk

C'est-à-dire que la jointure de l'ensemble des sous schémas redonne exactement le schéma initial ou encore que les tables des projections suffisent à reconstituer la table de départ.

Dans l'exemple écritPar cité plus haut, la décomposition est avec pertes. Car on ne connaît plus l'éditeur du livre.

Par contre la décomposition suivante :

R4(numéro_livre, auteur_nom, éditeur_nom)
R5(numéro_livre, titre_livre,nb_page)
R6(auteur_nom, auteur_adresse)
R7(éditeur_nom, éditeur_adresse)

est une décomposition sans perte de ecritPar(numéro_livre, titre_livre,nb_page, auteur_nom, auteur_adresse, éditeur_nom, éditeur_adresse)

Ceci provient du fait que le numéro du livre détermine son titre et son nombre de pages, le nom de l'auteur détermine son adresse ainsi que le nom de l'éditeur qui détermine son adresse également.

Lemme 1 :
Soit $U = X \cup Y \cup Z$ réunion d'ensembles disjoints d'attributs.
Soit R un schéma relationnel défini sur U, sur lequel existe la dépendance fonctionnelle X-> Y
alors la décomposition R1(X, Y) R2(Y, Z) est sans perte pour R.

iii. Test de décomposition sans perte d'information

Soit R un schéma relationnel, F un ensemble de dépendances fonctionnelles, R1,R2 ... Rk une décomposition.

Construire un tableau T ayant k lignes et n colonnes : La colonne j correspond à l'attribut Aj, la ligne i correspond au sous schéma Ri

Algorithme de poursuite:
Première phase :
Dans chaque case T[i,j] du tableau, on mettra ai si l'attribut Ai appartient à Ri et bij sinon.
Deuxième phase :
Pour chaque dépendance X-> Y de F, chaque fois que l'on trouve deux lignes ayant les mêmes valeurs (a ou b) pour les attributs de X on égalise les cases correspondant aux attributs de Y, en mettant prioritairement des ai.

Important : chaque fois que deux symboles sont égalisés il faut veiller à remplacer toutes les occurrences du symbole modifié.

Il faut répéter ces opérations jusqu'à ce que le tableau ne varie plus ou bien jusqu'à ce qu'apparaisse une ligne ne contenant que des ai : dans ce dernier cas seulement, la décomposition est sans perte.

Exemple 1
R(A,B,C,D)
F {AB→ C, A→D }
R1(A,B,C)
R2(A,D)

Au départ

R	A	B	C	D
R1	a_1	a_2	a_3	b_{14}
R2	a_1	b_{22}	b_{23}	a_4

AB→ C n'apporte aucune modification (il faudrait pour cela que $b_{22} = a_2$) auquel cas on changerait b_{23} en a_3
Par contre du fait de A → D

R	A	B	C	D
R1	a_1	a_2	a_3	a_4
R2	a_1	b_{22}	b_{23}	a_4

On a donc une ligne ne contenant que des ai et donc la décomposition est Sans Perte d'Information.
Remarque (dans cet exemple on pouvait appliquer le lemme vu plus haut puisque A est une clef de R2

exemple 2 :

R(A,B,C,D,E)

F {A ⟶ C, B ⟶ C, C ⟶ D, DE ⟶ C, CE ⟶ A}

R1(A,D)

R2(A,B)

R3(B,E)

R4(C,D,E)

R5(A,E)

Au départ on a donc

R	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{23}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{33}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{53}	b_{54}	a_5

En appliquant A ⟶ C

R	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{33}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{13}	b_{54}	a_5

En appliquant B ⟶ C

R	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{13}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{13}	b_{54}	a_5

En appliquant C ⟶ D

R	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	a_4	b_{25}
R_3	b_{31}	a_2	b_{13}	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	b_{13}	a_4	a_5

En appliquant DE $\rightarrow$ C

R	A	B	C	D	E
R_1	a_1	b_{12}	a_3	a_4	b_{15}
R_2	a_1	a_2	a_3	a_4	b_{25}
R_3	b_{31}	a_2	a_3	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	a_3	a_4	a_5

-- car tous les b_{13} se changent en a_3
En appliquant CE $\rightarrow$ A

R	A	B	C	D	E
R_1	a_1	b_{12}	a_3	a_4	b_{15}
R_2	a_1	a_2	a_3	a_4	b_{25}
R_3	a_1	a_2	a_3	a_4	a_5
R_4	a_1	b_{42}	a_3	a_4	a_5
R_5	a_1	b_{52}	a_3	a_4	a_5

Comme la ligne R_3 ne contient que des a_i la décomposition est SPI.

Exemple 3 :

On reprend l'exemple 2
En prenant
R(A,B,C,D,E)
F {A $\rightarrow$ C, B$\rightarrow$C, C$\rightarrow$D, DE$\rightarrow$C, CE$\rightarrow$A}
R_1(A,D)
R_2(A,B)
R_3(B,E)
R_4(C,D)
R_5(A,E)
Au départ on a donc

R	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{23}	b_{24}	b_{25}
R_3	b_{31}	a_2	b_{33}	b_{34}	a_5
R_4	b_{41}	b_{42}	a_3	a_4	b_{45}
R_5	a_1	b_{52}	b_{53}	b_{54}	a_5

Comme précédemment on obtient le tableau

R	A	B	C	D	E
R_1	a_1	b_{12}	b_{13}	a_4	b_{15}
R_2	a_1	a_2	b_{13}	a_4	b_{25}
R_3	b_{31}	a_2	b_{13}	a_4	a_5
R_4	b_{41}	b_{42}	a_3	a_4	b_{45}
R_5	a_1	b_{52}	b_{13}	a_4	a_5

Mais plus aucune DF n'apporte de modifications ! La décomposition n'est pas SPI !

Dans le cas d'un schéma R(X, Y, Z) décomposé en deux sous schémas R1(X, Y), R2(Y, Z) (on suppose X,Y et Z des ensembles d'attributs disjoints) la décomposition est sans perte si et seulement si Y -> X ou bien Y -> Z

3. Décompositions sans perte de dépendances

On choisit de stocker les données suivant une décomposition S du schéma
→ il faut vérifier que la base reste cohérente, i.e. qu'elle satisfait les df
→ à chaque m.a.j., il faut reconstruire la relation sur le schéma de départ U, sur lequel sont énoncées les df.

Problème : les jointures sont coûteuses
→ est-il possible de vérifier la cohérence de la base sans reconstruire la relation sur U, i.e. en se servant uniquement des relations stockées ?
→ oui si S « incorpore » un ensemble de df G équivalent à F

Exemple : Soit la relation **Examen(module, date, étudiant)** avec l'ensemble de dépendances F (module -> date, étudiant date -> module) signifiant qu'un module est associée à une date unique (examen par exemple) et qu'à une date donnée un étudiant ne peut passer qu'un seul module.

La décomposition en **R_1(module, date)** et **R_2(module, étudiant)** est sans perte d'information mais on ne peut vérifier que la dépendance **etudiant,date -> module** est bien respectée sauf à reconstituer la table initiale (ce que l'on souhaite éviter, en général)

On dit que cette dépendance a été perdue dans la décomposition. L'idée va donc consister à prendre des décompositions de sorte que l'on puisse vérifier un ensemble de dépendances (dans chaque sous schéma) suffisant pour être certain que toutes les dépendances de F sont bien vérifiées.

Soit une relation R et une décomposition R_1,R_2 ... R_n

Soit G_i la projection de F^+ sur R_i

Remarque : Gi = Gi+ et G, union des Gi est dans F+.

La décomposition sera sans perte de dépendance si G+ = F+

Remarque : si l'on note Fi la projection de F (et non de F+) sur Ri, Fi est inclus dans Gi mais ce n'est pas nécessairement réciproque.

Exemple

Soit R(A,B,C), F={A -> B, B -> C} et R1 (B,C) R2(A,C) (on remarquera, que la décomposition est bien SPI)
F1 = {B -> C} et F2 = {} mais F+ = {A -> B, B -> C, A -> C} + les DF triviales donc G1 = {B -> C} + les DF triviales et G2 = {A -> C} + les DF triviales. Il est clair que G (G1 U G2) ne contient pas F+ car il manque A -> B donc la décomposition n'est pas Sans Perte de Dépendance mais on voit bien que G2 contient strictement F2.

Remarque : une décomposition peut être SPD sans être SPI.

Exemple :
R(A,B,C,D)
F={ A -> B, C -> D}
R1(AB), R2(CD)

Il est clair que F= F1 U F2. Comme Gi contient Fi, G contient F donc F+. La décomposition est Sans Perte de Dépendance, pourtant elle n'est pas SPI car R1 inter R2 est vide et contredirait le lemme concernant la décomposition de 2 relations.