

7. Normalisation des Bases de Données relationnelles

7.1 Introduction

Nous avons appris auparavant à manipuler les bases de données relationnelles, nous allons nous intéresser maintenant à la manière d'élaborer une "base relationnelle".

Nous formulons l'hypothèse de départ que la phase d'analyse préalable nous a fourni un schéma relationnel "universel" : $\mathcal{R}$ défini sur un ensemble d'attributs U , plus un ensemble de prédicats sur la sémantique des attributs, plus un ensemble de contraintes d'intégrité.

7.2 Difficultés liées au caractère relationnel de la modélisation.

Avant de se pencher sur les "formes normales" reprenons les "pièges" auxquels expose la structure relationnelle.

Si l'on modélise mal on peut tomber sur des répétitions, des inaptitudes, des pertes d'informations.

Exemple : Nous disposons d'une modélisation relative au domaine bancaire :

$\mathcal{R}(\text{Nomagence}, \text{N}^\circ\text{Prêts}, \text{Client}, \text{Montant}, \text{Ch d'Affaires})$ correspondant au prédicat : "Une agence de nom Nomagence possède un chiffre d'affaires et consent des prêts de Numéro donné à des Clients pour un montant déterminé".

Les contraintes suivantes (dépendances fonctionnelles) sont établies :

Nomagence $\rightarrow$ Ch d'Affaires

Client $\text{N}^\circ\text{Prêt} \rightarrow$ Montant

Soit une instance r de $\mathcal{R}$:

r

Nomagence	N°Prêt	Client	Montant	Ch.d'Affaires
Montpellier	17	Dupont	1000	90 M
Montpellier	23	Durand	2000	90M
Béziers	15	Nestor	1500	45M
Nîmes	20	Adam	500	85M
Béziers	16	Achille	2000	45M
Nîmes	18	Désiré	1500	85M
....	...	...	...	

7.2.1 Divers problèmes

On veut insérer un nouveau client "Amanda" qui contracte le prêt 30 de montant 1500 auprès de l'agence de Béziers, il faut aussi rajouter l'information redondante relative au chiffre d'affaires 45M.

Si une agence change de nom tous les n-uplets relatifs à cette agence doivent subir la modification.

Si une nouvelle agence se crée sans encore aucun client comment insérer l'information sans valeur nulle? si on tolère les valeurs nulles les contraintes ne sont plus respectées.

7.2.2 Décomposition de schéma

Idée : Décomposer le schéma initial en sous-schémas, mais plusieurs décompositions sont envisageables.

7.2.2.1 Premier essai de décomposition

Soit une décomposition de $\mathcal{R}$:

$\mathcal{R}_1(\text{Nomagence, Ch d'Affaires})$ et $\mathcal{R}_2(\text{Nomagence, N°Prêt, Client, Montant})$

telle que, quelle que soit r instance de $\mathcal{R}$, r_1 instance de $\mathcal{R}_1$ et r_2 instance de $\mathcal{R}_2$:

$r_1 = \Pi \text{ Nomagence, Ch d'Affaires } (r)$

$r_2 = \Pi \text{ Nomagence, N°Prêt, Client, Montant } (r)$

Cette décomposition vérifie les propriétés suivantes :

- $r_1 \bowtie r_2 = r$ (jointure conservative)
- les contraintes sont toujours valables : Nomagence $\rightarrow$ Ch d'Affaires sur r_1 , Client N°Prêt $\rightarrow$ Montant sur r_2 (dépendances conservatives)

7.2.2.2 Second essai

On peut aussi envisager la décomposition de $\mathcal{R}$ en :

$\mathcal{R}_1(\text{Nomagence, N°Prêt, Montant})$, $\mathcal{R}_2(\text{Client, Montant})$ et $\mathcal{R}_3(\text{Nomagence, Ch d'Affaires})$

dont une occurrence est

r1			r2		r3	
Nomagence	N°Prêt	Montant	Client	Montant	Nomagence	Ch d'affaires
Montpellier	17	1000	Dupont	1000	Montpellier	90 M
Montpellier	23	2000	Durand	2000	Béziers	45 M
Béziers	15	1500	Nestor	1500	Nîmes	85 M
Nîmes	20	500	Adam	5000		
Béziers	16	2000	Achille	2000		
....	...	...	Désiré	500		
			...			

Divers problèmes :

Supposons que l'on souhaite poser la requête suivante :

Quelles sont les agences auprès desquelles Achille a emprunté ?

$\Pi \text{ Nomagence } (\sigma \text{ Client}='Achille' (r_2) \bowtie r_1)$ donne Béziers et **Montpellier** ! La jointure donne plus de n-uplets (jointure non conservative donc perte d'information).

De manière informelle nous voyons que cette deuxième décomposition n'est pas satisfaisante, dans la première décomposition l'intersection des ensembles d'attributs des schémas relationnels $\mathcal{R}_1$ et $\mathcal{R}_2$ donnait Nomagence clé de $\mathcal{R}_1$;

dans la deuxième décomposition l'intersection des ensembles d'attributs des schémas relationnels $\mathcal{R}_1$ et $\mathcal{R}_2$ donnait Montant or un client peut emprunter dans une même agence le même montant pour des prêts différents, ou bien emprunter le même montant dans des agences différentes ...etc.

Nous nous attacherons donc à décrire une méthodologie amenant à une **“bonne” décomposition**.

7.3 Normalisation sous dépendances fonctionnelles

7.3.1 Définitions

Les *dépendances fonctionnelles* constituent un type important de *contrainte d'intégrité* intervenant dans une structure relationnelle.

Un schéma relationnel $\mathcal{R}$ défini sur un ensemble d'attributs U vérifie la dépendance fonctionnelle $G \rightarrow D$ où G et D sont des sous-ensembles ou parties de U

$$\text{ssi } (\forall r \text{ instance de } \mathcal{R}) (\forall t, t' \in r \text{ tel que } t[G] = t'[G] \Rightarrow t[D] = t'[D])$$

(si deux n-uplets de r coïncident sur G alors ils coïncident sur D)

$t[G]$ note la projection de t sur G .

On dit que G détermine fonctionnellement ou identifie D , que D est fonctionnellement dépendant de G ou est identifié par G .

Remarque :

Si DG est l'ensemble des attributs de $\mathcal{R}$ on ne peut avoir deux n-uplets $t, t' \in r$ instance de $\mathcal{R}$ coïncidant sur G car sinon ils seraient égaux. Or les relations sont des ensembles, un n-uplet ne figure qu'une seule fois.

Exemple :

Soit le schéma relationnel EMPLOI_DU_TEMPS(Professeur,Heure,Salle,Classe,Matière) associé au prédicat intentionnel "tout enseignement d'une matière est effectué par un professeur pour une classe donnée dans une salle donnée à une heure donnée"

L'ensemble des dépendances fonctionnelles suivant peut lui être associé :

{Professeur $\rightarrow$ Matière ; Un professeur n'enseigne qu'une seule matière

Professeur Heure $\rightarrow$ Classe ; Un professeur à une heure donnée enseigne à une seule classe

Heure Salle $\rightarrow$ Professeur ; A une heure donnée et dans une salle donnée ne peut se trouver qu'un seul professeur

Heure Classe $\rightarrow$ Salle ; A une heure donnée et pour une classe donnée ne peut être associé qu'une seule salle}

Notons que selon l'observateur qui modélise l'ensemble de dépendances fonctionnelles peut être différent :

{Professeur $\rightarrow$ Matière ; Heure Classe $\rightarrow$ Professeur ; Heure salle $\rightarrow$ Classe}

Le problème sera de savoir si les ensembles de DF aboutissent au même schéma normalisé.

7.3.2 Equivalence entre ensembles de DF

7.3.2.1 Fermeture transitive d'un ensemble de DF

Définitions :

- *conséquence logique*

Etant donné un schéma relationnel $\mathcal{R}(U,F)$, une dépendance fonctionnelle f sera une conséquence logique de F (au sens des formules logiques) si f est vérifiée pour toute relation R de $\mathcal{R}$.

- *fermeture transitive d'un ensemble de DF*

Etant donné un ensemble F de DF, d'autres DF peuvent en être la conséquence logique. Comme les DF sont en nombre fini (si U est fini), l'ensemble des conséquences existe et s'appelle la fermeture transitive de F , notée F^+ .

exemple si (1) $X \rightarrow Y$ et (2) $Y \rightarrow Z$ sont 2 DF sur $\mathcal{R}$ alors $X \rightarrow Z$ est une DF sur $\mathcal{R}$.

en effet soit $\mathcal{R}$ contenant les attributs XYZ ,

si t et t' sont tels que $t[X] = t'[X]$ alors (1) $\Rightarrow t[Y] = t'[Y]$ et (2) $\Rightarrow t[Z] = t'[Z]$.

Il existe plusieurs moyens de dériver toutes les "conséquences" d'un ensemble de DF, nous en citerons un, celui utilisant des règles de dérivation appelées *axiomes d'ARMSTRONG*.

Règles principales

DF1 réflexivité	si $X \supseteq Y$ alors $X \rightarrow Y$
DF2 augmentation	si $X \rightarrow Y$ et Z est un ensemble d'attributs, alors $XZ \rightarrow YZ$
DF3 transitivité	si $X \rightarrow Y$ et $Y \rightarrow Z$ alors $X \rightarrow Z$

Nous admettons sans le démontrer que l'ensemble de ces règles de dérivation constituent un ensemble *valide* (sain ou consistant) et *complet*.

Définition :

Un système de règles de dérivation est *valide* (*sain* ou *consistant*), si lorsque f est dérivée de F à l'aide du système f est une conséquence logique de F (le système de règles ne permet donc d'engendrer que des DF valables).

Un système de règles de dérivation est *complet* si lorsque f est une conséquence logique de F alors f est dérivable de F en utilisant ce système de règles.

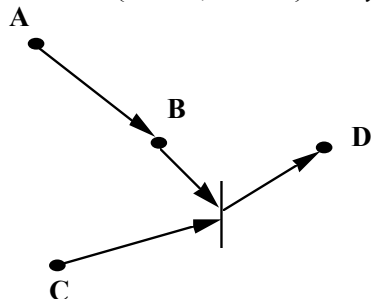
On utilise aussi des conséquences de ces règles qui sont utiles en pratique lors des manipulations de DF.

DF4 union	si $X \rightarrow Y$ et $X \rightarrow Z$ alors $X \rightarrow YZ$
DF5 décomposition	si $X \rightarrow YZ$ alors $X \rightarrow Y$ et $X \rightarrow Z$
DF6 pseudo-transitivité	si $X \rightarrow Y$ et $WY \rightarrow Z$ alors $WX \rightarrow Z$

7.3.2.2 Fermeture transitive d'un ensemble d'attributs par rapport à un ensemble de DF.**Définition :**

L'ensemble des attributs directement atteignables à partir de X par rapport à F est appelé fermeture transitive des attributs de X par rapport à F et notée X^+ .

exemple : Soit $F = \{ A \rightarrow B, BC \rightarrow D \}$ et l'*hypergraphe* correspondant



$A^+ / F = AB$
 $B^+ / F = B$
 $C^+ / F = C$
 $D^+ / F = D$
 $BC^+ / F = BCD$
 $AC^+ / F = ABCD$

Il y a un algorithme général simple pour calculer X^+ .

Algorithme Fermeture_trans(X, F, X^+)

{calcule dans X^+ la fermeture de X par rapport à F , ens de DF}

$X^+ := X$; ENU := F; {ENU ensemble des DF non utilisées}

tant que $\langle \exists \text{ une DF } G \rightarrow D \text{ dans ENU tel que } G \in X^+ \rangle$ **faire**

 choisir (et enlever) un arc $G \rightarrow D$ de ENU tel que $G \in X^+$

$X^+ := X^+ \cup \{D\}$;

Remarque : d'autres algorithmes plus sophistiqués ont été développés.

Définition

Soit F un ensemble de DF, F^+ la fermeture transitive de F est l'ensemble des DF :

$\{G \rightarrow D \text{ tq } D \in G^+ / F\}$

Exemple

Soit $F = \{ A \rightarrow B, BC \rightarrow D \}$

$F^+ = \{ A \rightarrow AB, B \rightarrow B, C \rightarrow C, D \rightarrow D, AB \rightarrow AB, AC \rightarrow ABCD, AD \rightarrow ABD, BC \rightarrow BCD, BD \rightarrow BD, CD \rightarrow CD, ABC \rightarrow ABCD, ABD \rightarrow ABD, ACD \rightarrow ABCD, BCD \rightarrow BCD, ABCD \rightarrow ABCD \}$

Théorème

Deux ensembles de dépendances fonctionnelles F et F' sont **équivalents**

ssi $F^+ = F'^+$

ssi $F^+ \subseteq F'^+$ et $F'^+ \subseteq F^+$

ssi $F \subseteq F'^+$ et $F' \subseteq F^+$

ssi toute DF de $F \in F'^+$ et toute DF de $F' \in F^+$

ssi pour toute DF de F $G \rightarrow D$, alors G^+ par rapport à F' contient D et

pour toute DF de F' $G \rightarrow D$, alors G^+ par rapport à F contient D

Grâce au théorème précédent, on ramène le calcul de la fermeture transitive de l'ensemble des dépendances fonctionnelles F à celui des **fermetures des parties gauches des dépendances fonctionnelles de F** .

On peut en effet montrer qu'en remplaçant $G \rightarrow D$ par $G \rightarrow G^+$, on a un ensemble de DF équivalent à F :

$\{F - (G \rightarrow D) \cup (G \rightarrow G^+)\}^+ = F^+$

7.3.2.3 Eliminer la redondance : obtention d'une Couverture irredondante Minimale (CIM)**a - minimisation des parties gauches de DF :**

définition : $A \in G$ est redondant dans $G \rightarrow D$ par rapport à F

ssi $D \in \{G - \{A\}\}^+$ par rapport à F

On appellera dépendance fonctionnelle **minimale** = une DF sans attribut redondant en partie gauche.

Exemple : $F = \{A \rightarrow B, AB \rightarrow C\}$; $AB \rightarrow C$ n'est pas minimale/ F , $A \rightarrow C$ est minimale !

b - redondance d'une DF ($G \rightarrow D$) par rapport à un ensemble F de DF

définition : $G \rightarrow D$ est redondant par rapport à F

ssi $D \in G^+$ par rapport à $(F - \{G \rightarrow D\})$

ssi $(F - \{G \rightarrow D\})^+ = F^+$

Pour chasser la redondance des dépendances fonctionnelles, rappelons qu'il faut décomposer $G \rightarrow D_1 D_2 \dots D_n$ en n dépendances fonctionnelles **élémentaires** (un seul attribut en partie droite)
 $G \rightarrow D_1, G \rightarrow D_2, \dots, G \rightarrow D_n$

Exemple : $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow BC\}$; $A \rightarrow BC$ n'est pas élémentaire : on la décompose en $A \rightarrow B$, déjà présente, et $A \rightarrow C$. Cette dernière est redondante car $A^+ / \{A \rightarrow B, B \rightarrow C\} = \{A, B, C\}$

c - couverture irredondante minimale (CIM) d'un ensemble F de DF

C'est un ensemble F' de DF équivalent à F qui ne contient aucune DF redondante, et dont les parties gauches sont minimales.

Pour en obtenir une (il n'y a pas forcément unicité), il faut appliquer les tests a et b dans cet ordre.

7.3.2.4 Clés de schéma relationnel**a - surclé K d'un schéma relationnel**

définition : K est surclé (ou identifiant) de $\mathcal{R}$ sur U ssi $K \rightarrow U$ est une DF sur $\mathcal{R}$

Deux n-uplets distincts ne peuvent avoir la même surclé (sinon ils coïncident entièrement et une réalisation R de $\mathcal{R}$ contient 2 tuples égaux)

U est toujours une surclé (triviale)

b - clé d'un schéma relationnel

définition : K est une clé ssi (K est une surclé de $\mathcal{R}$ et K est minimale (ne contient pas strictement une autre surclé K'))

ssi non $\exists A \in K$ tel que $K - \{A\} \rightarrow U$

c - calcul d'une clé

On essaie de minimiser la partie gauche de $U \rightarrow U$

Remarque : une clé sera représentée par un soulignement continu au niveau du schéma relationnel associé.
exemple : EMPLOI_DU_TEMPS(Professeur, Heure, Salle, Classe, Matière).

Algorithme clé (U, F, K)

{calcule une clé K d'un schéma relationnel $\mathcal{R}$ sur U avec un ensemble de DF F }

$K := U$; {surclé}

tant que $\exists$ un attribut $A \in K$ tel que $K - \{A\} \rightarrow U$ **faire**

choisir un tel attribut A ;
 $K := K - \{A\}$;

Il existe d'autres algorithmes capables de calculer l'ensemble des clés possibles pour un schéma $\mathcal{R}$.

7.3.3 Propriétés d'une décomposition

La normalisation va permettre de décomposer un schéma relationnel $\mathcal{R}(U, F)$ en un ensemble de schéma relationnels ayant une "bonne forme" incorporant l'ensemble des dépendances fonctionnelles F et réduisant la redondance des données et par suite réduisant les diverses difficultés de mise à jour citées initialement.

• On appelle décomposition d'un schéma $\mathcal{R}$ défini sur U en sous-schémas $\mathcal{R}_i$ définis sur U_i sous-ensemble de U ssi $U = \cup \text{des } U_i$ (recouvrement de l'ensemble des attributs)

Les décompositions effectuées doivent présenter un certain nombre de propriétés que nous allons détailler.

7.3.3.1 Décomposition sans perte

Définitions

On appelle **décomposition sans perte** d'un schéma $\mathcal{R}$ soumis à un ensemble de dépendances fonctionnelles F en sous-schémas $\mathcal{R}_i$ définis sur $\{U_i\}$ ssi pour toutes les instances R de $\mathcal{R}$ vérifiant F , on a :

$$R = \prod_{U_1} (R) \cup \prod_{U_2} (R) \dots \cup \prod_{U_n} (R)$$

7.3.3.1.1 Test d'une décomposition

Algorithme **Test_sans_perte**($\mathcal{R}, \{U_i\}, F$) : booléen

{retourne Vrai si $\{U_i\}$ est une décomposition sans perte}

• Construire un tableau T de n colonnes (n étant le cardinal de U) et de k lignes (k est le nombre de sous-schémas de la décomposition) : la ligne i correspond au schéma $\mathcal{R}_i$ et la colonne j correspond à l'attribut j de U .

• **Pour tous** les attributs A_j si $A_j \in \mathcal{R}_i$ mettre $T(i, j) = a_j$ sinon $T(i, j) = b_{i, j}$

• **Pour toute** DF $G \rightarrow D$ de F **faire**

Pour toutes les lignes qui sont identiques pour tous les attributs A_j de G **faire** coïncider les attributs de D de la manière suivante :

- si l'un d'entre eux est de la forme a_j , mettre les autres à a_j

- sinon les mettre égaux à l'un d'entre eux (bij par ex)

Résultat := si la table T possède au moins une ligne remplie de ai alors la décomposition est sans perte.

Exemple :

Soit $\mathcal{R}(\{A,B,C\}, \{A \rightarrow B, B \rightarrow C\})$ décomposé en $R_1(A,B), R_2(B,C)$;

Le tableau est initialisé à :

i\j	A	B	C
R1	a1	a2	b13
R2	b21	a2	a3

Le traitement de la première DF $A \rightarrow B$ ne donne rien ; en revanche, la seconde DF $B \rightarrow C$ modifie le tableau en :

i\j	A	B	C
R1	a1	a2	a3
R2	b21	a2	a3

La décomposition est donc sans perte.

7.3.3.1.2 Autre test

Les théorèmes suivants peuvent aussi permettre de vérifier qu'une décomposition est sans perte.

Théorème

Si $\mathcal{R}$ soumis à F (ens de DF) se décompose en $\{\mathcal{R}_1(U_1), \mathcal{R}_2(U_2)\}$
alors cette décomposition est sans perte par rapport à F ssi
 $(U_1 \cap U_2) \rightarrow (U_1 - U_2)$ ou $(U_1 \cap U_2) \rightarrow (U_2 - U_1)$

ssi les attributs communs sont clés de l'une ou l'autre des schémas relationnels décomposés.

Si $\mathcal{R}$ soumis à F (ens de DF) se décompose en $\{U_1, U_2, \dots, U_n\}$ sans perte,
et si U_1 soumis à $\pi_{U_1}(F)$ se décompose en $\{V_1, V_2\}$ sans perte,
alors $\{V_1, V_2, U_2, \dots, U_n\}$ est une décomposition sans perte de $\mathcal{R}$ soumis à F

Ce dernier théorème permet une décomposition sans perte par itérations successives.

7.3.3.2 Décomposition avec préservation des DF

Définition

La projection d'un ensemble de DF sur un ensemble d'attributs est l'ensemble des DF de F^+ qui ont tous leurs attributs dans Z

$$\pi_Z(F) = \{G \rightarrow D \mid G \rightarrow D \in F^+ \text{ et } G \cup D \subseteq Z\}$$

Une décomposition préserve un ensemble F de DF ssi $\{\cup \pi_{\mathcal{R}_i}(F)\}^+ = F^+$
ssi "l'union des projections est équivalent à l'ensemble initial"

Exemple

Soit $\mathcal{R}$ sur ABC soumis à $F = \{A \rightarrow B, B \rightarrow C\}$

a) la décomposition sur AC, BC préserve-t-elle les DF ?

F^+ sans les dépendances triviales = $\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$

$\pi_{AC}(F) = \{A \rightarrow C\}, \pi_{BC}(F) = \{B \rightarrow C\}$

$\pi_{AC}(F) \cup \pi_{BC}(F) = \{A \rightarrow C, B \rightarrow C\}$

et $A \rightarrow B$ n'est pas retrouvée donc la décomposition ne préserve pas les DF.

De plus, cette décomposition n'est pas sans perte : $AC \cap BC = C$ et C n'est clé d'aucune des 2 relations !

b) au contraire la décomposition AB, BC les préserve et est sans perte.

Il existe des méthodes pour vérifier la préservation des DF dans une décomposition mais nous ne les examinerons pas au niveau de ce cours.

7.3.4 Formes Normales

7.3.4.1 1ère FN

Un schéma relationnel $\mathcal{R}$ est en première forme normale si tous les attributs de U sont atomiques (ou monovalués).

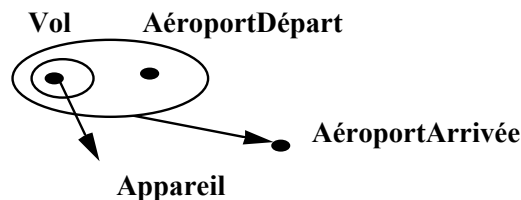
7.3.4.2 2ème FN

Un schéma relationnel $\mathcal{R}$ défini sur U et sur un ensemble de dépendances fonctionnelles F est en 2FN ssi tout attribut n'appartenant pas une clé dépend directement d'une clé.

Exemple :

Soit $\mathcal{R}(\text{Appareil}, \text{Vol}, \text{AéroportDépart}, \text{AéroportArrivée})$

avec $F = \{ \text{Vol} \rightarrow \text{Appareil}, \text{Vol AéroportDépart} \rightarrow \text{AéroportArrivée} \}$



L'unique clé de $\mathcal{R}$ est Vol AéroportDépart

Vol, AéroportDépart appartiennent à la clé, AéroportArrivée dépend directement de la clé, mais Appareil dépend d'une partie de la clé et non de la clé entière donc le schéma n'est pas en 2FN. Il faut scinder en deux relations : $R1(\underline{\text{Vol}}, \text{Appareil})$, $R2(\underline{\text{Vol}}, \underline{\text{AéroportDépart}}, \text{AéroportArrivée})$

7.3.4.3 3ème FN

Un schéma de relation $\mathcal{R}$ soumis à $F(\text{CIM})$ est en 3ième forme normale ssi pour toute DF $X \rightarrow A$ de F :

- soit X est une clé,
- soit A appartient à une clé.

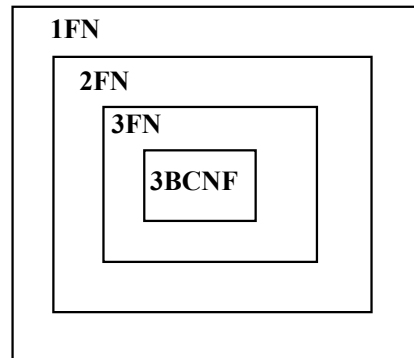
Par exemple, soit $\mathcal{R}$ sur ABC soumis à $F = \{ A \rightarrow B, B \rightarrow AC \}$

Alors $\{R1(\underline{A}, B), R2(\underline{B}, C)\}$ est en 3FN

Ainsi que $\{R1(\underline{B}, A), R2(\underline{B}, C)\}$ et $\{R1(\underline{B}, A), R2(\underline{A}, C)\}$

7.3.4.4 3ème BCNF (forme normale de Boyce-Codd)

$\mathcal{R}$ soumis à $F(\text{CIM})$, est en BCNF ssi pour toute DF $G \rightarrow D$ de F, G est une clé.



Remarquons que pour l'exemple précédent, $\{R1(\underline{B},A,C), R2(\underline{A},B)\}$ est en 3BCNF.

7.3.5 Décomposition en 3FN

Théorème :

Tout schéma relationnel peut décomposer en schémas sous 3ième forme normale, sans perte et avec préservation des Dépendances fonctionnelles.

Il existe divers algorithmes de décomposition (nous n'en étudierons qu'un "classique" dû à Bernstein).

Algorithme de décomposition en 3 ième forme normale

L'algorithme prend en entrée un schéma $\mathcal{R}$ soumis à F et donne en résultat un ensemble $\{\mathcal{R}_i(U_i, F_i)\}$ tel que tout $\mathcal{R}_i$ est en 3FN. Le schéma relationnel $\mathcal{R}$ et surtout F sont eux-mêmes le résultat d'une phase de modélisation.

1. Calculer une CIM (F1) à partir de l'ensemble de toutes les dépendances fonctionnelles élémentaires de F en chassant la redondance des attributs de partie gauche, puis les dépendances fonctionnelles redondantes.
2. • Trouver les clés de F1
3. • Regrouper les DF de F1 ayant même partie gauche $G_j \rightarrow D_j$. Fabriquer les schémas de relation maximaux par l'inclusion ensembliste.

$$\mathcal{R}_i \{U_i = G_j \cup D_j ; G_j \rightarrow D_j\}$$

4. • Si aucune clé ne figure dans une des relations produites à l'étape précédente ajouter un schéma de relation

$$\mathcal{R}_i = \{K ; K \rightarrow K\}$$

Remarque : Cet algorithme peut conduire à une décomposition en 3BCNF (meilleur des cas).

7.3.6 Normalisation par rapport à d'autres contraintes

Dans certains cas de modélisation, les dépendances fonctionnelles ne suffisent pas à traduire toute la sémantique de l'application. Il faut donc introduire de nouvelles contraintes et les rajouter dans le processus de normalisation.

7.3.6.1 Dépendances de jointure

Définition

Le schéma relationnel $\mathcal{R}(U)$ admet la **dépendance de jointure** notée

* $[U_1, U_2, \dots, U_i]$ où chaque U_i est un sous-ensemble de U

ssi $\mathcal{R}_1(U_1) \bowtie \mathcal{R}_2(U_2) \dots \bowtie \mathcal{R}_i(U_i)$. c'est à dire ssi toute relation R extension de $\mathcal{R}(U)$ est la jointure naturelle des relations $R_1, R_2, \dots, R_i$ extensions des schémas $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_i$.

Chaque schéma relationnel $\mathcal{R}_i(U_i)$ est appelé **composante de jointure**.

Un n-uplet u ne peut appartenir à R que si chacune de ses projections $u[U_i]$ est un n-uplet de R_i $R[U_i] \in R_i(U_i)$

Exemple : Soit le schéma de relation VINS(Buveur,Cru,Producteur) modélisant des vins bus par des buveurs d'un cru donné et commandés à un producteur produisant ce cru et une de ces extensions :

BUVEUR	CRU	PRODUCTEUR
Jacques	Chablis	Pierre
Jacques	Chablis	Nicolas
Jacques	Volnay	Nicolas
Louis	Chablis	Nicolas

Cette extension présente encore des redondances.

Si la contrainte d'intégrité : " tout buveur ayant bu un cru et ayant commandé à un producteur produisant ce cru, a aussi commandé ce cru à ce producteur" est déclarée, le schéma relationnel VINS obéit à la dépendance de jointure :

* [Buveur Cru, Buveur Producteur, Cru Producteur]
et peut être décomposé en trois schémas (composantes de jointure).

BUVEUR	CRU	BUVEUR	PRODUCTEUR	CRU	PRODUCTEUR
Jacques	Chablis	Jacques	Pierre	Chablis	Pierre
Jacques	Volnay	Jacques	Nicolas	Chablis	Nicolas
Louis	Chablis	Louis	Nicolas	Volnay	Nicolas

Dans le processus de normalisation vu précédemment, on rajoute les DJ en introduisant chaque ensemble d'attributs associé à chaque composante de jointure CJ (comme une DF $U_i \rightarrow$).

7.3.6.2 Dépendances d'inclusion

Définition

Soit deux schémas relationnels $\mathcal{R}_1$ sur U_1 et $\mathcal{R}_2$ sur U_2
soit X_1 et X_2 deux attributs tels que $X_1 \in U_1$ et $X_2 \in U_2$

On a une dépendance d'inclusion (notée $\in_i$) $X_1 \in_i X_2$ ssi
 $\forall R_1, R_2$ extensions ou instances de $\mathcal{R}_1, \mathcal{R}_2$ on a
 $\forall t_1 \in R_1 \exists$ un n-uplet $t_2 \in R_2$ tel que $t_2[X_2] = t_1[X_1]$

Remarque : généralisation à des ensembles d'attributs, on note $X_1 X_2 \in_\beta Y_1 Y_2$

$X_1 X_2$ est appelé la partie gauche ou membre inclus, et $Y_1 Y_2$ la partie droite ou membre incluant de la DI.

attention : $X_1 X_2 \in_\beta Y_1 Y_2 \Rightarrow X_1 \in_\beta Y_1$ et $X_2 \in_\beta Y_2$ mais l'inverse est faux.

Propriétés

L'existence de DF et de DI implique l'existence d'autres DF et DI que l'on peut obtenir grâce aux règles suivantes :

DI1 : réflexivité $X \in_\alpha X$

DI2 : transitivité si $X \in_\alpha Y$ et $Y \in_\beta Z$ alors $X \in_\gamma Z$ avec $\gamma = \alpha\beta$.

DI3: héritage de DF : si $XL \in_\alpha GD$ et $G \rightarrow D$ alors $X \rightarrow L$

L'introduction de DI dans le processus de normalisation peut entraîner des suppressions d'attributs, des simplifications par fusion de schémas.