Chapitre III : La normalisation

I. Introduction :

Objectif : Déterminer un « bon » schéma relationnel qui décrit l'application.

Décrire des produits et leurs fournisseurs.

Schéma 1 :

Produit (<u>N°P</u>, NomP, Poids, couleurP)
Fournisseur (<u>NF</u>, NomF, AdrF, TelF)
Livraison (<u>N°P, NF</u>, Quté, Date)

Si aucune livraison ➔ on ne connaît pas le téléphone du fournisseur

Schéma 2 :

Produit (<u>N°P</u>, NomP, Poids, couleurP)
Fournisseur (<u>NF</u>, NomF, AdrF)
Livraison (<u>N°P, NF</u>, TelF, Quté, Date)

Si n livraison ➔ n*le numéro de téléphone ➔ redondance

Pour décrire qu'une relation est de bonne qualité, il existe le processus de normalisation.

X,Y,Z, désignent soit un attribut, soit un groupe.

A,B,C désignent un attribut.

II. Dépendance fonctionnelle (DF):

<u>Définition :</u> Etant donné une relation R(X,Y,Z) ; il existe une DF de **Y** vers **Z** ($Y \rightarrow Z$), si étant donnée 2 tuples quelconque de **R,** s'ils ont même valeur pour **Y** alors ils ont nécessairement même valeur pour **Z**.

<u>Définition :</u> Une DF X → B est une DF élémentaire, si B est attribut unique et si X est un ensemble minimum d'attributs, c'est-à-dire :

Il n'existe pas X' C X tel que X'→ B

DF élémentaires :

NP → NomP
NP → Poid
NP → Couleur
NomP → NP
NomP → Poid
NomP → Couleur

Propriétés de DF :

1. Réflexivité : Si Y est un sous-ensemble de X alors X → Y
2. Augmentation : Si X → Y alors XZ → YZ
3. Transitivité : Si X → Y et Y → Z alors X → Z (DF déduite)
4. Autodétermination : X → X
5. Décomposition : X → YZ alors X → Y et X → Z
6. Union : Si X → Y et X → Z alors X → YZ
7. Composition : Si X → Y et Z → A alors XZ → YA

Exple :

$$\left.\begin{array}{l}A\\B\end{array}\right| \rightarrow C \rightarrow D \text{ alors } AB \rightarrow D \text{ est déduite}$$

$$\left.\begin{array}{l}\left.\begin{array}{l}A\\B\end{array}\right| \rightarrow C\\ \qquad\quad D\end{array}\right| \rightarrow E \text{ alors } ABC \rightarrow E \text{ est déduite}$$

<u>Définition :</u> Etant donnée une relation et un ensemble F de DF portant sur les attributs de cette relation, on appelle graphe minimum des DF de la relation, tout ensemble de DF élémentaires non déduites équivalent à F.

III. Décomposition d'une relation :

<u>Objectif :</u> Etant donné une relation non satisfaisante, trouver un ensemble clé de relations satisfaisantes qui décrivent les mêmes informations.

<u>Méthode :</u> Consiste à décomposer la relation initial en 2 (ou +) relations.

<u>Définition :</u> Une relation X,Y,Z est décomposable sans perte d'informations en R1(X,Y) et R2(X,Z), si on peut reconstruire R à partir de R1 et R2.

<u>Théorème de **Heath** :</u> Toute relations R(X,Y,Z) est décomposable sans perte d'information en R1(X,Y) et R2(X,Z), s'il y a dans R une DF X → Y

IV. 1ére et 2éme Forme normal (FN) :

Définition : Une relation est en 1ére FN si chaque valeur de chaque attribut de chaque tuple est une valeur simple.

Fournisseur (NF, NomF, Addr, Tel, NomProduit, Prix)
└> Redondance

Fournisseur (NF, NomF, Addr, Tel)
Catalogue (NF, NomProduit, Prix)
└>Cette décomposition est sans perte d'information (Théorème d'Heath), sans perte de DF.

Méthode à appliquer :

1. Vérifier que la relation est en 1ére FN
2. Etablir le graphe minimum des DF
3. Déterminer tous les identifiants : Si la relation est mal normalisé, on décompose à l'aide du graphe en relation mieux normaliser.
4. Déterminer la FN si la relation est mal normalisée, décomposé à l'aide d'un graphe, en relation mieux normalisée.

Définition : Une relation est en 2éme FN, si elle est en 1ére FN et si chaque attribut qui ne fait pas partie d'un identifiant mais dépend d'un identifiant.

V. 3éme FN (Eliminer les DF déduite) : (Dépend directement id.)

Fournisseur (NF, Pays, Ville)

Hypothèse : pas d'hypothèse dans les villes

NF → Ville
Ville → Pays
NF → Pays (déduite)

NF → Ville → Pays

FN de fournisseur 2éme FN
Pb de redondance, à cause de Ville, Pays.

NF → NomF
NF → Adr
NF → Tel
NF, NomProduit → Prix

← DF élémentaire déduite

Graphe minimum DF

1éme Source : Fournisseur (NF, NomF, Adr, Tel) ; 2éme FN
↑ il faut qu'il appartient
2éme Source : Catalogue (NF, NomProduit, Prix) ; 2éme FN

On décompose pour éviter ce problème :

FR (NF, Ville) 3^{éme} FN — Sans perte de DF
Géo (Ville, Pays) 3^{éme} FN — Sans perte d'information (Th. Heath)

Définition : Une relation est en 3^{éme} FN si elle est en 1ère FN et si chaque attributs qui ne fait pas partie d'un identifiant dépend d'un identifiant directement.

[On peut toujours décomposer en 3^{éme} FN sans perte de DF et d'information]

VI. FN de Boyce-Codd (kart) :

Catalogue (NF, NomF, NomProduit, Prix) 1ère FN, 2^{éme}FN, 3^{éme} FN

Hypothèse : Pas d'homogène chez le fournisseurs

2 id. Qui sont (NF, NomProduit) ou (NomF, NomProduit)

Problème de redondance entre NF et NomF

27/09/06

Prix ← NF ↔ NomF, NomProduit

Prix ← NomF ↔ NF, NomProduit

Si une relation est en 1^{ére} et 3^{éme} FN et si chaque attribut qui ne fait pas partie d'un id. dépend d'un id. directement.

Source ↑ NF, Autre source ↑ NomF

Fournisseur (NF, NomF) ; 2 id. 3^{éme} FN & BC
Catalogue2 (NF, NomProduit, Prix) 1 id. 3^{éme} FN & BC

Sans perte d'information ni de DF

Plus de redondance entre NF et NomF

Définition : Une relation est en FN de BC si elle est en 1^{ére} FN et si toute source de DF est un id.

[1^{ére}, 2^{éme}, 3^{éme} FN Pas en BC car NF → NomF]

Autre exemple :

Place (N°Etudiant, Matière, Rang) 3^{éme} FN & BC
Représente le rang obtenu pour chaque étudiant dans chaque matière

Hypothèse : pas d'ex-acque

2 id.s = (N°Etudiant + Matière) ou (Matière + Rang)

<u>N°Etudiant Matière</u> On ne décompose pas

Rang

3^{éme} Exemple :

Enseignement (<u>N°Etudiant, Matière, Prof</u>) 3^{éme} FN, pas en BC car prof → Matière

Hypothèse : Chaque prof enseigne une seul matière et que pour chaque matière, chaque étudiant suit les cours d'un seul prof.

2 id. = (N°Etudiant + Matière) ou (N°Etudiant + Prof)

<u>N°Etudiant Matière</u>

Prof

N°Etudiant	Matière	Prof	
1	BD	Savonnet	
~~1~~	~~S&R~~	~~Savonnet~~	Refusé car id. (N°Etudiant + Prof)
~~1~~	~~BD~~	~~Leclerq~~	Refusé car id. (N°Etudiant + Matière)
2	S&R	Savonnet	Accépté

On est sans perte d'information, mais on a perdu la DF (N°Etudiant, Matière) → Prof

- Pb car non respect de la DF Prof → Matière
- Donc on décompose (par le th. De Heath)
- R1(Prof, Matiere) 3^{éme} FN, BC
- R2 (Prof, N°Etudiant) 3éme FN, BC

↓

Conclusion : On reste en 3^{éme} FN avec la relation enseignement. BC cas des relations avec plusieurs id. qui se chevauchent.

VII. Méthode de décomposition :

a) Décomposition en Boyce-Codd sans perte d'information :

Il est possible de décomposer toute la relation en FN de BC sans perte d'information en appliquant récursivement le théorème de Heath.

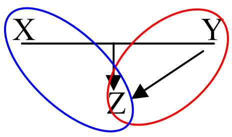
Tant que de 1 des relations obtenues par décomposition R (X, Y, Z), il existe une DF X → Z, on décompose R en R1 (X, Z) et R2 (X, Y)

- On peut avoir de pertes de DF (cf. exemple)
- On peut être amené à <u>trop</u> de décomposition
  └> requête affiche à écrire

b) Décomposition en 3$^{\text{éme}}$ FN sans perte d'information ni de DF :

Il est possible de décomposer toute relation R en 3$^{\text{éme}}$ FN sans perte d'information et de DF suivant la méthode çi-dessous.

- Créer pour chaque source de DF (une relation) comprenant un attribut. Toutes les cibles direct : X ↔ Y
- Eventuellement une relation supplémentaire constituer d'un attribut composant un id. de R.



VIII. Dépendances multivaluées :

Catalogue (N°Article, Taille, Couleur) 3$^{\text{éme}}$ FN BC

Décrit la section d'habillement d'un magasin de vente par correspondance.

Quantité	Taille	Couleur
1	38	Noir
1	40	Noir
1	42	Noir
1	44	Noir
1	38	Bleu
1	40	Bleu
1	42	Bleu
1	44	Bleu

Taille et couleurs sont indépendant

Redondance,

Un id. = N°Article + Couleur

Par contre, il existe un autre type de dépendance. Chaque article existe pour chaque taille dans toutes les couleurs et chaque article existe pour chaque couleur dans toutes les tailles. C'est la notion de dépendance multivaluées.

Définition : Soit une relation R(X, Y, Z), Z pouvant être vide, on dit que Y dépend de façon multivaluée de X, on note X →>

Si lors que l'on a des tuples <x,y,z>et<x',y',z'>€ R ; alors <x, y',z>et<x',y,z'>€ R

Employé (N°E, Nom, Prénom, poste, salaire, diplôme, annédipl) ;

DF : N°E → Nom
N°E → Prénom
N°E → poste
N°E → salaire

On veut stocker la liste de ses diplômes ; dépendance multivaluées
N°E →> (diplôme, annédipl)

IX. Théorème :

Une relation R(X, Y, Z) est décomposable sans perte d'info en R1(X, Y) et R2(X, Z), s'il y a une DF multivaluée X →> Y tel qu'il existe dans R d'autre attributs en plus R(X, Y, Z) a pour source d'un id. de R.

CatalogueBC (N°Art, Couleur, Taille)
└> CatCouleur (N°Art, Couleur)
└> CatTaille (N°Art, Taille)

EmployeDipl (N°E, NomDipl, DateObt)

X. Dependance multivaluée de jointure 5éme FN:

Antibiotique (Maladie, Germe, Antibiotique) ;

M1	G1	A1
M1	G1	A3
M1	G2	A3
M2	G3	A3
M2	G1	A2

03/10/06

Il existe une DM car « si un germe G est présent dans une maladie M, si un antibiotique A agit sur le genre G et si l'antibiotique A agit sur la maladie M alors le triplet <M, G, A> est présent dans la relation ».

XI. Conclusion :

Une décomposition est bonne si :
- ✓ elle est utile (on doit generer des relations mieux normalisées)
- ✓ pas de perte d'information
- ✓ pas de perte de DF, DM

LMD : Langage de Manipulations de Données

1) Pourquoi a-t-on besoin d'un langage spécifique ?

└> La structure d'une relation est un tableau à 2 dimensions ; n'importe quel langage de programmation sait manipuler des tableaux.

└> Inconvénient d'un langage de programmation est que les programmes des utilisateurs doivent connaître l'organisation des tuples dans les relations pour accéder aux informations.

➔ Indépendance pg/données n'est plus vérifiée.

Les LMD doivent être purement conceptuels.
i.e porter sur les notions de relation e attribut et domaine et ignorer tout de l'organisation interne des relations.

↓

Les LMD fournissent un ensemble réduits de fonctions limitées à celles que le SGBD est capable d'optimiser mais suffisamment vaste pour répondre à toutes les questions.

2 grandes familles :

SQL {
- Lg algébrique → algèbre relationnel
- Lg prédicatif → calcul des tuples

↓

Norme