

CHAPITRE 5

Normalisation d'une relation

1. Introduction

Objectif : Définir un bon schéma relationnel qui décrit bien l'entreprise.

Exemple : on veut décrire les produits et leurs fournisseurs. On peut le faire avec le schéma suivant, schéma 1 (on suppose que chaque produit est d'une couleur unique) :

Produit (NP, NomP, Couleur, Poids)
Fournisseur (NF, NomF, Adr, Tel)
Livraison (NP, NF, Date, Qté)

Autre schéma proposé pour le même sujet, schéma 2 :

Produit (NP, NomP, Couleur, Poids)
Fournisseur (NF, NomF, Adr)
Livraisonbis (NP, NF, Date, TêLF, Qté)

Exemples de problèmes rencontrés lors des mises à jour de la base de données décrite par le schéma 2 :

S'il n'y a plus de livraison pour un fournisseur son numéro de téléphone est perdu. S'il existe N livraisons pour un fournisseur, le numéro TêLF est répété N fois, il faut vérifier que c'est le même. Pour insérer une nouvelle livraison, il faut enregistrer à nouveau ce numéro TêLF. Ces problèmes n'existent pas avec le schéma 1 qui est meilleur que ce second schéma.

Le processus de transformation d'une relation posant des problèmes lors des mises à jour en relations n'ayant pas ces problèmes, est appelé processus de normalisation.

On mesure la qualité d'une relation (ou sa capacité à représenter le monde réel sans générer les problèmes aperçus ci-dessus) par son degré de normalisation. Une relation peut être (de la moins bonne à la meilleure) en 1ère forme normale, en 2ème forme normale, en 3ème forme normale, en forme normale de Boyce Codd, en 4ème forme normale ... (chaque forme normale implique les précédentes).

Notation : pour la suite, les lettres X, Y, Z ... désigneront soit des attributs, soit des ensembles d'attributs; les lettres A, B, C, ... désigneront des attributs.

2. Dépendances fonctionnelles et graphe des dépendances fonctionnelles

Définition :

Etant donné une relation, $R(X, Y, Z)$, il existe une **dépendance fonctionnelle**, ou "DF", de Y vers Z, notée $Y \rightarrow Z$, si :

étant donné deux tuples quelconques de R, s'ils ont même valeur pour Y, alors ils ont nécessairement même valeur pour Z.

On appelle Y source de la DF, et Z cible de la DF.

Exemple : pour la relation Produit (NP, NomP, Poids, Couleur), il y a les DF suivantes en supposant qu'il n'existe pas deux produits de même nom :

NP \rightarrow NomP, NomP \rightarrow NP, NP \rightarrow Poids
 NomP \rightarrow Poids, NP \rightarrow Couleur, NomP \rightarrow Couleur
 NP \rightarrow (NomP, Poids, Couleur)
 (NP, NomP) \rightarrow Poids, (NP, NomP) \rightarrow Couleur

Définition : une DF, $X \rightarrow B$, est une **dépendance fonctionnelle élémentaire** si B est un attribut unique, et si X est un ensemble minimum d'attributs (ou un attribut unique).

Exemples : Dans la relation Produit, les DF :

NP \rightarrow (couleur, poids) et (NP, NomP) \rightarrow Poids
 ne sont pas élémentaires.

Mais les DF :

NP \rightarrow Couleur, NP \rightarrow Poids, NP \rightarrow NomP
 NomP \rightarrow Couleur, NomP \rightarrow Poids, NomP \rightarrow NP
 sont élémentaires.

La DF :

(NP, NF, date) \rightarrow Qté
 de la relation Livraison est élémentaire.

Chaque DF traduit un fait du monde réel. Les DF élémentaires traduisent des faits atomiques. La DF, NP \rightarrow Couleur, signifie que chaque produit, identifié par son numéro, est d'une seule couleur. La DF, (NP, NF, date) \rightarrow Qté, signifie qu'un même fournisseur ne peut livrer plusieurs fois le même jour le même produit avec des quantités différentes.

Propriété des dépendances fonctionnelles :

Si dans une relation, on a les deux dépendances fonctionnelles, $X \rightarrow Y$ et $Y \rightarrow Z$, alors on a aussi la dépendance fonctionnelle $X \rightarrow Z$ qui est dite **déduite** des deux autres.

Définition :

Etant donné une relation et un ensemble \mathbb{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 \mathbb{F} en ce sens que toute DF de \mathbb{F} peut être déduite des DF du graphe.

Une méthode pour savoir si une DF, $X \rightarrow Y$, est déduite des autres est la suivante:

- établir un graphe (non minimum) des DF,
- supprimer la DF $X \rightarrow Y$ du graphe,
- parcourir tous les chemins possibles partant de X et suivant les DF. La DF, $X \rightarrow Y$, est déduite si un (ou plusieurs) de ces chemins atteint Y.

Le graphe minimum des DF sert essentiellement à définir des relations normalisées.

Tout graphe (minimum ou pas) des DF peut être employé pour la recherche des identifiants des relations de la façon suivante:

chercher, sur le graphe des DF de la relation, tout ensemble minimum d'attributs, X, tel que tous les chemins partant de X et suivant les DF atteignent tous les autres attributs du graphe. Alors X est un identifiant de la relation.

3. Décomposition d'une relation

Objectif : étant donné une relation non satisfaisante, en ce sens qu'elle implique des répétitions au niveau de sa population et qu'elle pose des problèmes lors des insertions/modifications/suppressions de tuples (voir la relation Livraisonbis ci-dessus), trouver un ensemble de relations satisfaisantes et qui décrivent les mêmes informations.

La méthode consiste à décomposer la relation en deux (ou plusieurs) relations. Pour cela, il est nécessaire de disposer de deux opérations qui permettent, l'une, de découper une relation en sous-relations (la projection), et l'autre, de recomposer la relation à partir de ses sous-relations (la jointure). Ces deux opérations sont définies ci-dessous.

Définition : Projection d'une relation:

étant donné une relation R (A1, A2,...,An) et un sous-ensemble de ses attributs Ai1, Ai2,..., Aip, la projection de R sur Ai1, Ai2,..., Aip:

$$\pi[Ai1, Ai2, \dots, Aip] R$$

créé une nouvelle relation d'attributs Ai1, Ai2,..., Aip, et de population égale à l'ensemble des tuples de R tronqués à Ai1, Ai2,..., Aip (sans doubles); c'est-à-dire formellement:

$$\{t / \exists r (r \in R \wedge t.Ai1 = r.Ai1 \wedge t.Ai2 = r.Ai2 \wedge \dots \wedge t.Aip = r.Aip)\}$$

Exemple sur la base FormaPerm (voir chapitre précédent):

On veut les noms des cours qui ont un prérequis. Pour cela, on fait la projection de la relation Prérequis sur l'attribut nomC (on dit que prérequis est projeté sur nomC):

$$\pi[nomC] \text{Prérequis}$$

nomC
système
BD
SI

Les valeurs doubles (système, SI) ont été éliminées.

Définition : Jointure naturelle de deux relations:

étant donné deux relations ayant au moins un attribut commun, R1(X, Y) et R2(Y, Z), la jointure naturelle de R1 et R2 s'indique de la façon suivante (une autre notation utilise un "papillon" à la place de l'étoile):

$$R1 * R2.$$

C'est une nouvelle relation d'attributs (X,Y,Z), et dont la population est constituée de l'ensemble des tuples de R1 concaténés à ceux de R2 qui ont même valeur pour Y; c'est-à-dire formellement:

$$\{t / \exists t1 \in R1, \exists t2 \in R2 (t.X=t1.X \wedge t.Y=t1.Y \wedge t.Z=t2.Z \wedge t1.Y=t2.Y)\}$$

Attention: les tuples de R1 (ou de R2) qui n'ont pas de correspondant dans R2 (R1) n'apparaissent pas dans le résultat de la jointure.

Exemple sur la base de données FormaPerm:

Si on veut les renseignements sur les étudiants qui ont réussi (au moins) un cours, il faut faire une jointure de Personne, Etudiant et Obtenu:

Personne * Etudiant * Obtenu

n°P	nom	adr	dateN	n°E	nomC	note	année
6666	Walter	Lausanne	3/3/78	22	algo	10	1998

6666	Walter	Lausanne	3/3/78	22	C	9	1998
5555	Bernard	Yverdon	2/2/78	44	algo	9	1999

Les attributs sur lesquels on a réalisé la jointure sont n°P pour Personne*Etudiant et n°E pour Etudiant*Obtenu. Un étudiant a été éliminé (le n°E 36) parce que cet étudiant n'a réussi aucun cours. On peut décomposer une relation en un ensemble de relations projetées, si à partir des relations projetées, on peut, en faisant leur jointure, retrouver la relation initiale. Ainsi toute requête, qu'elle soit posée à la relation initiale ou aux relations issues de la décomposition, donnera le même résultat. La relation initiale et les relations issues de la décomposition constituent deux bases de données équivalentes.

Définition: Une décomposition d'une relation $R(X,Y,Z)$ en deux relations $R1=\pi[X,Y]R$ et $R2=\pi[X,Z]R$ (obtenues par projection de la relation R respectivement sur les attributs X,Y et X,Z) est dite "sans perte d'information" si $R = R1 * R2$.

Exemples sur la base FormaPerm:

- La décomposition de la relation Personne en:
 $\pi[n^{\circ}P, \text{nom}] \text{ Personne}$ et $\pi[n^{\circ}P, \text{adr}] \text{ Personne}$
est une décomposition sans perte d'information, car:
 $(\pi[n^{\circ}P, \text{nom}] \text{ Personne}) * (\pi[n^{\circ}P, \text{adr}] \text{ Personne}) = \text{Personne}$.
On remarquera cependant que cette décomposition est inutile, car la relation Personne est "bonne".

- La décomposition de la relation Personne en:
 $\pi[n^{\circ}P, \text{nom}] \text{ Personne}$ et $\pi[\text{nom}, \text{adr}] \text{ Personne}$
est une décomposition avec perte d'information, car:
 $(\pi[n^{\circ}P, \text{nom}] \text{ Personne}) * (\pi[\text{nom}, \text{adr}] \text{ Personne}) =$

n°P	nom	adr
1111	Rochat	Lausanne
1111	Rochat	Renens
6666	Walter	Lausanne
5555	Bernard	Yverdon
2222	Rochat	Lausanne
2222	Rochat	Renens
3333	Muller	Morges

La jointure sur l'attribut nom a multiplié les tuples Rochat. On remarquera que l'on parle de "perte d'information" alors qu'il y a plus de tuples qu'auparavant. En effet, on ne sait plus quels sont les "bons" tuples et les "faux" tuples. Les décompositions sans perte d'information assurent que toute requête posée à la base de données initiale (avant décomposition) et la requête équivalente posée sur la base de données issue de la décomposition donnent le même résultat. Mais on verra plus loin (paragraphes 6 et 7) que ces décompositions sans perte d'informations peuvent cependant perdre des dépendances fonctionnelles et poser des problèmes lors des mises à jour.

Le théorème suivant permet de savoir quand une décomposition est sans perte d'information.

Théorème de Heath:

Toute relation $R(X,Y,Z)$ est décomposable sans perte d'information en $R1=\pi[X,Y]R$ et $R2=\pi[X,Z]R$ s'il y a dans R une dépendance fonctionnelle de X vers Y ($X \rightarrow Y$).

Principe de la démonstration:

- $R1 * R2$ contient au moins tous les tuples de R , puisque tout tuple xyz de R crée un tuple xy dans $R1$ et un tuple xz dans $R2$, qui sont concaténés par la jointure en xyz .

- $R1 * R2$ ne peut pas contenir de tuples en plus de ceux de R . Ceci est démontré par l'absurde.

Soit xyz un tuple de $R1 * R2$, qui n'appartient pas à R .

xyz provient de deux tuples de R : xyz' et $xy'z$.

Etant donné que xyz n'appartient pas à R , on a: $z' \neq z$ et $y' \neq y$, ce qui est contraire à la DF $X \rightarrow Y$.

4. Première et deuxième formes normales

Normaliser une relation qui pose des problèmes de mise-à-jour (relation non normalisée), consiste à décomposer cette relation en relations sans problèmes (relations normalisées). La méthode à suivre est la suivante:

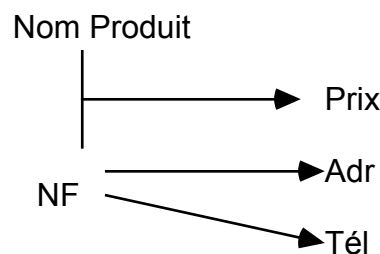
- 1/ vérifier que la relation est en première forme normale (voir définition ci-dessous);
- 2/ établir son graphe minimum des dépendances;
- 3/ déterminer, à l'aide du graphe, tous ses identifiants;
- 4/ déterminer, à l'aide du graphe, sa forme normale (voir définitions ci-dessous);
- 5/ si la relation n'est pas normalisée, décomposer, à l'aide du graphe, la relation en relations mieux normalisées (voir § 7.2).

Définition : Une relation est en **première forme normale** si chaque valeur de chaque attribut de chaque tuple est une valeur simple (tous les attributs sont simples et monovalués).

Soit une relation qui est en 1^{ère} forme normale, mais qui n'est pas en 2^{ème} forme normale:

Fournisseur1 (NF, NomProduit, Adr, Tel, Prix)

Graphe minimum des DF de Fournisseur1:



Une telle relation pose des problèmes :

Redondances : s'il existe 100 produits pour un fournisseur on va répéter 100 fois le nom, l'adresse, le téléphone du fournisseur.

Problème de mise à jour pour les insertions : quand on veut rajouter un produit, il faut rentrer à nouveau l'adresse et le téléphone du fournisseur.

Problème pour les suppressions : si on supprime (momentanément) la liste des produits d'un fournisseur, alors on supprime aussi le fournisseur.

Problème de mise à jour des tuples : si un fournisseur change d'adresse ou de téléphone, il faut faire cette mise à jour sur tous les 100 tuples !

Ces problèmes sont dus au fait que la relation n'est pas en 2^{ème} forme normale.

On décompose Fournisseur1 en deux relations de la façon suivante : pour chaque source de DF, on crée une relation ayant pour attributs la source et tous les attributs en dépendance fonctionnelle directe de cette source, en s'assurant qu'une (au moins) des deux sources est entièrement contenue dans les attributs communs aux deux relations ainsi créées (cf théorème de Heath). On obtient ainsi :

Fournisseur (NF, Adr, Tel)

Catalogue (NF, NomProduit, Prix)

qui sont en 2^{ème} forme normale (et même plus).

Cette décomposition est :

- sans perte d'information (NF est l'identifiant de la relation Fournisseur)

- sans perte de dépendance fonctionnelle (les DF sont soit dans l'une, soit dans l'autre des deux relations décomposées).

Définition : Une relation est en **deuxième forme normale** si elle est en première forme normale et si chaque attribut qui ne fait partie d'aucun identifiant dépend de tout identifiant entier (et non pas d'un morceau d'identifiant).

5. Troisième forme normale

Soit une relation qui est en deuxième forme normale :
Fournisseur2 (N°F, Pays, Ville)

Il existe les DF :

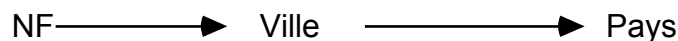
$NF \rightarrow Ville$ et $Ville \rightarrow Pays$

car on suppose qu'on n'a dans la base de données que des grandes villes sans homonymes. La DF :

$NF \rightarrow Pays$

est une DF déduite.

Graphe minimum des DF de Fournisseur2 :



Dans la relation Fournisseur2, il y a redondance : le pays d'une ville est répété, ce qui cause des problèmes de mise à jour. On décompose donc en :

Fourn (NF, Ville)

Géo (Ville, Pays)

Cette décomposition est sans perte d'information (Ville est identifiant pour Géo), et sans perte de DF (les DF non déduites sont soit dans Fourn, soit dans Géo).

Définition : une relation est en **troisième forme normale** si elle est en première forme normale et si chaque attribut qui ne fait partie d'aucun identifiant dépend uniquement des identifiants entiers.

NB : On peut toujours décomposer en troisième forme normale sans perte ni d'information, ni de DF; ce qui n'est pas vrai des formes normales plus poussées. D'où l'intérêt de cette troisième forme normale.

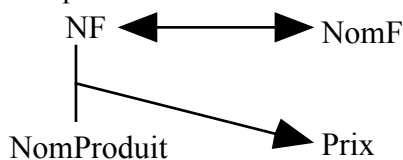
6. Relations à plusieurs identifiants: forme normale de Boyce Codd

Soit une relation qui est en troisième forme normale (on suppose qu'il n'y a pas d'homonyme chez les fournisseurs) :

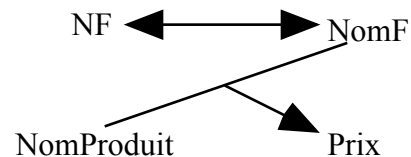
Catalogue3 (NF, NomF, NomProduit, Prix)

Identifiants : (NF + NomProduit) , (NomF + NomProduit)

Graphe minimum des DF de Catalogue3 :



ou



Dans la relation Catalogue3, il y a redondance entre NF et NomF, ce qui génère des problèmes lors des mises à jour. Par exemple si un fournisseur change de nom, il faut mettre à jour son nom dans tous les tuples correspondants à ses produits.

On décompose donc Catalogue3 en deux relations de forme normale de Boyce Codd, par exemple en:

Fournisseur (NF , NomF) d'identifiants : NF et NomF

Catalogue (NF , NomProduit , Prix) d'identifiant : (NF + NomProduit)

Définition :

Une relation est en **forme normale de Boyce Codd** si elle est en première forme normale et si toute source complète de DF est un identifiant entier.

La relation Catalogue3 n'est pas en forme normale de Boyce Codd parce que l'attribut NF est source complète de DF ($NF \rightarrow NomF$) et n'est pas un identifiant entier. Il en est de même pour NomF.

On remarque qu'une relation en forme normale de Boyce Codd, est en troisième forme normale.

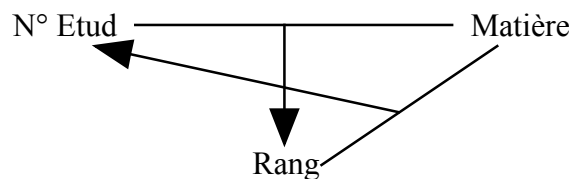
Autre exemple

La relation :

Place (N°Etud, Matière, Rang)

représente le rang obtenu par chaque étudiant pour chaque matière. On suppose qu'il n'y a pas d'ex aequo. Il y a deux identifiants : (N°Etud + Matière), (Matière + Rang).

Graphe minimum des DF de Place :



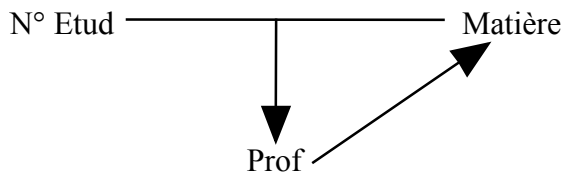
Est-on en troisième forme normale ? Oui, car il n'y a pas d'attribut qui ne fasse pas partie d'un identifiant. On est aussi en Boyce Codd. On ne décompose donc pas.

Autre exemple

Enseignement (N°Etud, Matière, Prof)

On suppose que chaque professeur enseigne une seule matière, et que pour chaque matière, chaque étudiant suit les cours d'un seul professeur.

Graphe minimum des DF de Enseignement :



La relation Enseignement a deux identifiants : (N°Etud + Matière) et (N°Etud + Prof). Elle est en 3e forme normale puisque tout attribut fait partie d'un identifiant. Mais elle n'est pas en forme normale de Boyce Codd puisque l'attribut Prof est source de DF et ne constitue pas un identifiant entier.

Cette relation présente des inconvénients (redondances, problèmes de mise à jour) dus au fait que la DF, $Prof \rightarrow Matière$, n'est pas traduite par la relation. Notamment si un professeur change de spécialité, il faut penser à faire cette mise à jour dans tous les tuples où apparaît ce professeur.

Cependant on ne peut pas décomposer la relation Enseignement sans perdre de DF. Par exemple, la décomposition selon Heath:

(Prof, Matière)

(N°Etud, Prof)

perd la DF: $(N°Etud, Matière) \rightarrow Prof$. Cette décomposition permet d'insérer le fait qu'un même étudiant suit deux cours portant sur la même matière avec deux professeurs différents, ce qu'interdisait la relation Enseignement.

Il n'y a pas de solution idéale dans ce cas. Pratiquement on conserve la relation initiale et on rajoute une contrainte d'intégrité spécifiant le fait qu'un professeur ne peut enseigner qu'une seule matière.

7. Méthodes de décomposition

Pour décomposer une relation non normalisée en un ensemble de relations normalisées, plusieurs méthodes conduisant à des algorithmes ont été proposées, mais aucune d'entre elles, appliquée de façon purement automatique, n'est totalement satisfaisante. En effet, elles peuvent proposer en résultat des relations normalisées, mais qui ne sont pas sémantiquement significatives.

7.1. Décomposition en Boyce Codd sans perte d'information

Il est possible de décomposer toute relation en forme normale de Boyce Codd sans perte d'information en appliquant récursivement le théorème de Heath :

Tant que dans une des relations obtenues par décomposition, $R(X,Y,Z)$, il existe une DF $X \rightarrow Z$, on décompose R en $\pi[X,Y]R$ et $\pi[X,Z]R$.

Cette méthode présente plusieurs inconvénients :

- elle peut conduire à trop décomposer, notamment en décomposant une relation qui est déjà en Boyce Codd.

Une décomposition n'est utile que si la relation initiale n'est pas assez normalisée et que les relations issues de la décomposition sont plus normalisées. Un schéma trop décomposé complique l'emploi de la base de données, car les requêtes seront plus complexes (nombreuses jointures).

- elle peut perdre des DFs, voir par exemple au paragraphe précédent le cas de la relation:

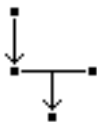
Enseignement (N°Etud, Matière, Prof).

Les dépendances perdues doivent alors être ajoutées au schéma décomposé sous la forme de contraintes d'intégrité.

7.2. Décomposition en troisième forme normale sans perte d'information ni de dépendance fonctionnelle

Il est possible de décomposer toute relation R en troisième forme normale sans perte d'information ni de DF, en suivant la méthode ci-dessous :

- créer pour chaque source de DF une relation comprenant comme attributs la source et toutes les cibles de cette source;
- si aucun des identifiants de R n'est présent dans au moins une des relations créées, ajouter une relation supplémentaire, constituée des attributs composant un des identifiants de R . Cette relation supplémentaire supprimera les tuples excédentaires lors de la jointure des relations issues de la décomposition pour reconstituer la relation initiale. Par exemple, le cas suivant le cas suivant décomposé en deux relations (sans relation contenant l'identifiant de la relation initiale), ne satisfait pas le théorème de Heath et génère lors de la jointure des tuples excédentaires.



Cette méthode de décomposition a l'inconvénient de générer parfois des décompositions redondantes, par exemple dans le cas de la relation Enseignement (N°Etud, Matière, Prof).



Il faut donc ensuite supprimer du résultat les relations qui sont incluses dans (c'est-à-dire égales à une projection d') une autre relation.

8. Dépendances multi-ensembles

Certaines relations sont en forme normale de Boyce Codd et contiennent cependant des redondances d'informations qui créent des problèmes lors de leurs mises à jour. Un exemple d'une telle relation, est la relation CatalogueBC :

CatalogueBC (N°Article, Taille, Couleur)

qui décrit la section habillement du catalogue d'un grand magasin de vente par correspondance. On suppose que Taille et Couleur sont indépendants.

CatalogueBC	N°Article	Taille	Couleur
	Pull	38	Bleu
	Pull	40	Bleu
	Pull	42	Bleu
	Pull	44	Bleu
	Pull	38	Vert
	Pull	40	Vert
	Pull	42	Vert
	Pull	44	Vert

Cette relation n'a pas de DF. L'identifiant est (N°Article + Taille + Couleur).

On est en Boyce Codd puisqu'il n'y a pas de DF (la proposition "toute source de DF est un identifiant entier" est vraie).

Par contre, il y a un autre type de dépendance: chaque article existe pour chaque taille dans toutes les couleurs, et réciproquement chaque article existe pour chaque couleur dans toutes les tailles. De telles dépendances sont appelées dépendances multi-ensembles (ou dépendances multivaluées).

Définition :

Soit une relation $R(X, Y, Z)$, il y a **dépendance multi-ensemble** (DM) $X \twoheadrightarrow Y$ si à toute valeur de X correspond un ensemble de valeurs de Y qui est totalement indépendant de Z .

Formellement, si $t1$ et $t2$ sont deux tuples de R , tels que $t1[X] = t2[X]$, alors il existe deux tuples $v1$ et $v2$ de R tels que :

$$\begin{aligned} v1[X] &= v2[X] = t1[X] \\ v1[Y] &= t1[Y], v1[Z] = t2[Z] \\ v2[Y] &= t2[Y], v2[Z] = t1[Z] \end{aligned}$$

On remarque que les DF sont des cas particuliers de DM.

Attention, une DM est dite élémentaire si sa source est minimale, mais sa cible n'est pas nécessairement constituée d'un seul attribut (contrairement aux DF).

Exemple: La relation suivante décrit le personnel d'une entreprise avec pour chaque employé, son nom, ses prénoms, son poste, son salaire et la liste de ses diplômes avec pour chacun l'année d'obtention.

Employé (n°E, nom, prénom, poste, salaire, diplôme, annéedipl)

Cette relation comprend des redondances dues aux dépendances multi-ensembles. Les dépendances, fonctionnelles et multi-ensembles, sont les suivantes :

$n^{\circ}E \rightarrow \text{nom}$, $n^{\circ}E \rightarrow \text{poste}$, $n^{\circ}E \rightarrow \text{salaire}$
 $n^{\circ}E \twoheadrightarrow \text{prénom}$, $n^{\circ}E \twoheadrightarrow (\text{diplôme}, \text{annéedipl})$

Mais il n'y a pas les DM de $n^{\circ}E$ vers diplôme ou annéedipl:

$n^{\circ}E \not\twoheadrightarrow \text{diplôme}$, ni $n^{\circ}E \not\twoheadrightarrow \text{annéedipl}$

car les deux attributs, diplôme et annéedipl, sont liés.

On remarque que chaque DM correspond, dans le modèle entité association, à deux attributs multivalués indépendants l'un de l'autre.

Propriété des dépendances multi-ensembles :

Si dans une relation $R(X, Y, Z)$, on a la dépendance multi-ensemble $X \twoheadrightarrow Y$, alors on a aussi la dépendance multi-ensemble $X \twoheadrightarrow Z$.

9. Quatrième forme normale

Comme les DF, les DM qui ne sont pas exprimées par une relation, créent des redondances et donc des problèmes lors des mises à jour. Par exemple, la création d'un nouvel article dans la relation CatalogueBC nécessite d'insérer $n \times m$ tuples, n étant le nombre de tailles et m celui des couleurs. La solution, comme avec les DF, est de décomposer, si possible, la relation sans perte d'information ni de dépendance.

La relation CatalogueBC qui contient deux DM :

$N^{\circ}\text{Article} \twoheadrightarrow \text{Taille}$ et $N^{\circ}\text{Article} \twoheadrightarrow \text{Couleur}$,

est donc décomposée de la façon suivante: création d'une relation par DM (ce qui revient en terminologie entité association à créer une relation par attribut multivalué).

ArticleCouleur ($N^{\circ}\text{Article}$, Couleur)

ArticleTaille ($N^{\circ}\text{Article}$, Taille)

Ces deux relations ne posent plus de problème lors des mises à jour.

Le théorème de Heath sur les décompositions sans perte d'information est généralisé au cas des dépendances multi-ensembles.

Théorème :

Toute relation $R(X, Y, Z)$ est décomposable sans perte d'information en $\pi[X, Y]R$ et $\pi[X, Z]R$, s'il y a une dépendance multi-ensemble de X vers Y ($X \twoheadrightarrow Y$).

Un inconvénient des DM, que n'ont pas les DF, est que leur existence dépend des valeurs prises par les autres attributs de la relation. Ainsi, lors des décompositions d'une relation, il faut, dans chaque relation décomposée, vérifier quelles sont les DM, car de nouvelles DM peuvent apparaître.

Définition :

Une relation R est en **quatrième forme normale** si elle est en première forme normale, et si toute dépendance fonctionnelle ou multi-ensemble de R ($X \rightarrow Y$ ou $X \twoheadrightarrow Y$), telle qu'il existe dans R d'autres attributs en plus de X et Y , a pour source un identifiant entier de R .

Cette définition de quatrième forme normale implique celle de Boyce Codd.

La relation CatalogueBC (de même qu'Employé) n'est pas en quatrième forme normale. Mais les deux relations issues de la décomposition de CatalogueBC, ArticleTaille et ArticleCouleur, le sont.

10. Conclusion

10.1. Rappel: bonne décomposition

Une décomposition en plusieurs relations résultat est bonne si :

1) elle est utile:

- elle crée des relations décomposées plus normalisées que la relation initiale,
- certaines DF ne doivent pas générer de décomposition: celles entre des attributs qui ne sont pas (ou très rarement) modifiés et qui sémantiquement ne méritent pas d'être décrites par une relation. Par exemple, il est inutile de décomposer la relation suivante qui décrit les viticulteurs:

Viticulteur ($n^{\circ}\text{viticulteur}$, nom, prénom, rue, village, codeNPA, téléphone, surface)
même s'il existe la DF : village \rightarrow codeNPA.

2) il n'y a pas de perte d'information lors de la décomposition :

- l'attribut (ou les attributs) commun entre les relations issues de la décomposition est, au moins pour l'une d'entre elles, source de DF ou de DM pour tous les autres attributs de cette relation.

3) il n'y a pas de perte de dépendance lors de la décomposition :

- toutes les dépendances non déduites doivent être dans l'ensemble des relations issues de la décomposition.

10.2. Deux méthodes de conception d'un schéma relationnel normalisé

1) Concevoir un schéma entité association, puis le traduire en relationnel.

Si le schéma entité association était "bon", alors la plupart des relations sont normalisées. Seules les relations qui proviennent d'attributs complexes contenant des dépendances peuvent ne pas l'être. Il suffit donc de vérifier la forme normale de ces relations. Un algorithme de traduction est présenté ci-dessous.

2) Créer une relation regroupant tous les attributs (appelée "relation universelle"). Etablir un graphe minimum des dépendances. Puis employer l'une ou l'autre des méthodes de décomposition vues au paragraphe 7.

10.3. Traduction d'un schéma entité association en relationnel

Il est possible de traduire un schéma entité association en un schéma relationnel en appliquant l'algorithme ci-dessous. Si le schéma entité association est bien construit, alors le résultat de la traduction sera un schéma relationnel normalisé en quatrième forme normale, sauf dans certains cas de dépendance à l'intérieur d'un attribut complexe. Une amélioration de l'algorithme consisterait à prendre en compte ces dépendances lors de la traduction des attributs complexes afin de ne générer que des relations en quatrième forme normale. Avec l'algorithme donné ci-dessous, il faut vérifier la forme normale de chaque relation obtenue à partir d'un attribut complexe.

Algorithme de traduction :

Pour chaque type d'entité, créer une relation de :

- nom = nom du type d'entité;
- attributs = attributs monovalués du type d'entité (pour les attributs complexes, prendre leurs attributs composants simples, avec pour nom le nom de l'attribut complexe concaténé à celui de l'attribut composant); les attributs facultatifs deviennent des attributs à valeur nulle possible.
Si le type d'entité est sous-type d'un autre type E, alors il faut ajouter l'(un des) identifiant(s) de E aux attributs de la relation. Soit IE cet (ensemble d') attribut(s);
- identifiants = ceux du type d'entité qui ne font pas intervenir d'attribut multivalué. Si tous les identifiants comprennent un attribut multivalué ou s'il n'y a pas d'identifiant, alors demander à l'administrateur de la base d'ajouter à la relation un nouvel attribut identifiant.
Si le type d'entité est sous-type d'un autre type E, alors IE est aussi un identifiant de la relation;
- identifiants externes = si le type d'entité est sous-type d'un autre type E, alors IE est un identifiant externe qui référence la relation E, sinon il n'y a pas d'identifiant externe.

Pour chaque attribut multivalué A d'un objet O (type d'entité, type d'association ou attribut multivalué complexe), créer une relation de :

- nom = nom de l'objet O, concaténé à celui de l'attribut A;
- attributs = l'attribut A (ou ses attributs composants simples et monovalués si A est complexe) + l'identifiant I (ou un des identifiants) de l'objet O + si la collection est de type liste un attribut ordre (de type Entier) qui définit l'ordre des valeurs + si la collection est de type multi-ensemble un attribut nombre (de type Entier) qui définit le nombre de valeurs identiques;
- identifiants = si la collection est de type liste, l'identifiant de la relation est constitué de l'identifiant I de l'objet O et de l'attribut ordre qui a été ajouté; sinon l'identifiant de la relation est constitué de l'identifiant I de l'objet O + si A est un attribut simple alors A, sinon un (des) attribut composant de A (choix à faire par l'administrateur de la base);
- identifiant externe = l'identifiant I de l'objet O.

Pour chaque type d'association dont au moins un rôle a pour cardinalité maximum 1,

soit R le type d'association liant :

E1 avec le rôle r1 et les cardinalités: x1,1

/* si R a plusieurs rôles de cardinalité maximum 1, choisir de préférence pour E1/r1 celui (un de ceux) de cardinalités 1,1 */

E2 avec le rôle r2 et les cardinalités: x2,y2

.....

En avec le rôle r1 et les cardinalités: xn,yn

alors ajouter à la relation décrivant le type d'entité E1 :

- aux attributs : l'identifiant (un des) de chacun des Ei liés sauf E1/r1, avec pour nom celui de Ei (ou celui de Ei concaténé au nom du rôle ri en cas d'homonymie), et les attributs monovalués de R (pour les attributs complexes, prendre leurs attributs composants simples, avec pour nom le nom de l'attribut complexe concaténé à celui de l'attribut composant); tous ces attributs ajoutés peuvent prendre la valeur nulle si le rôle r1 est facultatif (x1=0)
- aux identifiants externes : chaque identifiant, Ei (ou Ei.ri), ajouté à la relation est un identifiant externe qui référence la relation décrivant le type d'entité Ei.

Pour chaque type d'association dont tous les rôles ont une cardinalité maximum supérieure à 1,

soit R le type d'association liant :

E1 avec le rôle r1 et les cardinalités : x1,y1 y1>1

E2 avec le rôle r2 et les cardinalités : x2,y2 y2>1

.....

En avec le rôle r1 et les cardinalités : xn,yn yn>1

alors créer une relation de:

- nom = nom de R (en cas d'homonyme rajouter un numéro d'ordre);
- attributs = l'identifiant (un des) de chacun des Ei liés, avec pour nom celui de Ei (ou celui de Ei concaténé au nom du rôle ri en cas d'homonymie), et les attributs monovalués de R (pour les attributs complexes, prendre leurs attributs composants simples, avec pour nom le nom de l'attribut complexe concaténé à celui de l'attribut composant);
- identifiants = ceux de R;
- identifiants externes : chaque identifiant, Ei (ou Ei.ri), ajouté à la relation est un identifiant externe qui référence la relation décrivant le type d'entité Ei.

Contraintes

Certaines contraintes exprimées dans le schéma entité association peuvent être directement exprimées dans le schéma relationnel issu de la traduction. C'est le cas par exemple, des attributs obligatoires (de cardinalité minimale 1) qui deviennent des attributs NOT NULL en relationnel. Les contraintes qui ne peuvent pas être directement exprimées dans le schéma relationnel, seront traduites par des CHECK, ASSERTIONS ou TRIGGERS du SQL.

Exercices - Série 7

Exercice 1

Pour chaque relation ci-dessous:

- identifier les redondances éventuelles dans sa population,
- établir le (un) graphe minimum de ses dépendances,
- définir son (ses) identifiant(s),
- définir sa forme normale et la justifier,
- si nécessaire, proposer une décomposition optimale.

a. Pièce: description des pièces employées dans un atelier de montage.

Pièce (N°pièce, prix-unit, TVA, libellé, catégorie)

avec les dépendances fonctionnelles suivantes:

N°pièce → prix-unit, TVA, libellé, catégorie

catégorie → TVA

b. Prime: liste des primes attribuées au personnel technique en fonction du type de machine sur lequel il travaille

Prime (N°type-machine, nom-machine, N°techn, montant-prime, nom-techn)

avec les dépendances fonctionnelles suivantes:

N°type-machine → nom-machine

N°techn → nom-techn

(N°type-machine, N°techn) → montant-prime

c. Employé: description d'un employé travaillant sur un projet d'un laboratoire.

Employé (N°Emp, N°Lab, N°Proj, NomEmp, NomProj, adresse)

avec les dépendances fonctionnelles suivantes:

(N°Emp, N°Lab) → N°Proj

N°Emp → NomEmp

N°Emp → adresse

N°Proj → NomProj

d. Adresse

Adresse (rue, ville, code-postal)

avec les dépendances fonctionnelles suivantes:

code-postal → ville

(rue, ville) → code-postal

Exercice 2

Soit R la relation suivante, avec les dépendances:

$R(A, B, C, D, E, F, G)$

$AB \rightarrow C, AB \rightarrow D, AB \rightarrow E, AB \rightarrow F, B \rightarrow C, D \rightarrow E, D \rightarrow F, G \rightarrow A$

a. Etablir le (un) graphe minimum de dépendances. Quel(s) est (sont) l'identifiant(s) de R ?

- b. Quelle est la forme normale de R? Justifier votre réponse.
- c. Proposer une décomposition optimale de R.

Exercice 3: Université

On veut décrire les séances de travaux dirigés (TD) des unités de valeurs (UV) d'un département d'une université par la relation suivante:

FAC (N°TD, salle, horaire, N°enseignant, nom-enseignant, prénom-enseignant, N°UV, nomUV, N°étudiant, nom-étudiant, prénom-étudiant, adresse-étud., date-ins.)

L'enseignement, dans ce département est divisé en unités de valeurs, chacune étant identifiée par un numéro ou par son nom.

Un étudiant s'inscrit à une ou plusieurs UV (six au maximum), et pour chaque UV à un groupe de TD (N°TD). Les inscriptions dans les différentes UV sont indépendantes les unes des autres. On mémorise la date d'inscription de chaque étudiant à chaque UV (date-ins.).

Il y a une séance de TD par semaine pour chaque UV. Chaque TD a lieu dans une salle donnée et à un horaire donné. Les groupes de TD sont numérotés 1, 2, 3, ... pour chaque UV.

Un enseignant assure un ou plusieurs groupes de TD d'une ou plusieurs UV. Un groupe de TD d'une UV est assuré toute l'année par le même enseignant, plusieurs enseignants pouvant se partager les différents groupes de TD d'une même UV. Exceptionnellement, par manque d'enseignants, un même enseignant peut assurer simultanément deux TD différents situés dans deux salles contiguës.

On ne conserve que le prénom usuel de chaque personne.

- a. Quelles redondances et anomalies de mise à jour sont impliquées par cette relation FAC?
- b. Établir un graphe minimal des dépendances fonctionnelles de FAC. Quel est son (ses) identifiant(s)? Quelle est sa forme normale?
- c. Proposer une bonne décomposition de FAC et préciser la forme normale des nouvelles relations.

Exercices - Série 8

Exercice 1

Pour chaque relation ci-dessous:

- identifier les redondances éventuelles dans sa population,
- établir le (un) graphe minimum de ses dépendances,
- définir son (ses) identifiant(s),
- définir sa forme normale,
- si nécessaire, proposer une décomposition optimale.

a. Employé1: description du fait que chaque employé peut avoir plusieurs aptitudes (taper à la machine, cuisiner, ...) et savoir plusieurs langues.

Employé2 (N°emp, aptitude, langue)

b. Employé2: description du fait que chaque employé possède certaines aptitudes relatives à certains pays; par exemple, Alfred sait cuisiner des recettes italiennes et chinoises, et il sait lire le chinois; Chantal cuisine des recettes japonaises.

Employé3 (N°emp, aptitude, pays)

Exercice 2 : Association sportive

L'association sportive de l'université désire connaître les sports qu'aimeraient pratiquer les étudiants. De manière à choisir les horaires pour ces sports, on a demandé aux étudiants de donner la liste des horaires (jour et heure) auxquels ils sont libres et celle des sports qu'ils veulent pratiquer. La relation suivante a ainsi été créée :

Ass-Sport (N°étudiant , horaire , sport)

décrivant le fait que cet horaire est l'une des plages où l'étudiant est libre et que ce sport est l'un de ceux qu'il désire pratiquer.

Existe-t-il des redondances et anomalies dans cette relation? Lesquelles? Quel est le graphe des dépendances de cette relation? Quels sont ses identifiants et quelle est sa forme normale? Peut-on la décomposer? Comment?

Exercice 3

Soit la relation R avec les tuples suivants:

A	B	C
1	2	4
1	3	4
2	5	7
1	2	7
1	3	7
1	5	4
1	5	7

- Quelles sont les dépendances fonctionnelles ou multivaluées compatibles avec la population de la relation R ?
- R est-elle en quatrième forme normale? Sinon, décomposer la relation R en quatrième forme normale.

Exercice 4 : Les lignes de bus

Dans la base de données d'une société de transport, la relation suivante décrit pour les jours du mois en cours les affectations des conducteurs de bus aux lignes:

Affectation (N°ligne, N°conducteur, jour)

Plus précisément, cette relation signifie que tel jour, tel conducteur est affecté à telle ligne de bus.

Considérons indépendamment les unes des autres les contraintes potentielles suivantes:

- Un conducteur ne peut pas travailler sur deux lignes différentes le même jour;
- Un conducteur fait les mêmes lignes tous les jours où il travaille;
- Un conducteur ne fait qu'une seule ligne;
- Une ligne est toujours parcourue par les mêmes conducteurs;
- Une ligne n'est parcourue que par un seul conducteur;
- Un conducteur ne fait pas la même ligne deux jours différents.

7.1. Pour chacune de ces contraintes, définir la (ou les) dépendance fonctionnelle ou multivaluée qui lui correspond(ent).

7.2. Pour chacun des ensembles de dépendances suivants, dessiner le graphe minimum des dépendances de la relation Affectation, préciser quel est (sont) l'identifiant de la relation, quelle est sa forme normale (en le justifiant en une ligne). Si la relation Affectation n'est pas bien normalisée proposer une décomposition en précisant les identifiants des relations obtenues, leur forme normale et s'il y a eu perte de dépendance. Dans ce dernier cas, définir en français la (les) contrainte d'intégrité que la base devra satisfaire.

- L'ensemble des dépendances représentant les contraintes (a) et (c);
- L'ensemble des dépendances représentant les contraintes (c) et (e);
- L'ensemble des dépendances représentant la contrainte (d);
- L'ensemble des dépendances représentant les contraintes (a) et (f).