Chapitre 2

LE MODELE RELATIONNEL

Introduction

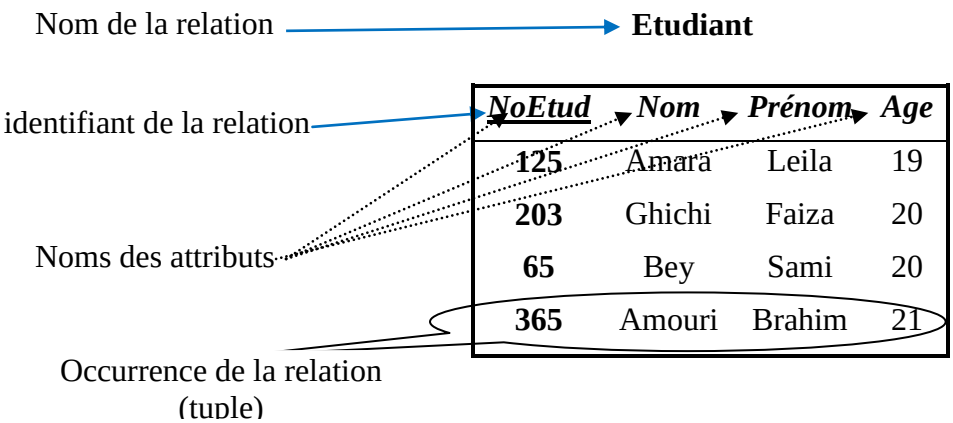
Le modèle relationnel a été inventé en 1970 et a fait l'objet de très nombreuses recherches qui ont débouché sur la réalisation et la commercialisation de SGBDs relationnels. C'est le modèle le plus utilisé par les SGBDs actuellement disponibles sur le marché.

C'est un modèle de données plus simple que celui de l'entité-association, ce qui explique son succès tant sur le plan théorique (théorie de la normalisation, définition des langages de manipulation de données), que sur le plan des réalisations. Mais cette simplicité est aussi sa faiblesse principale : c'est un outil trop pauvre sémantiquement pour pouvoir bien représenter la complexité du monde réel. C'est pourquoi d'autres modèles de type entité-association ou orientés objets ont été proposés.

1) Définitions

Les objets et associations du monde réel sont représentés par un concept unique : la relation.

- **Une relation** est un tableau à deux dimensions, souvent appelé **table**.
-



NoEtud	*Nom*	*Prénom*	*Age*
125	Amara	Leila	19
203	Ghichi	Faiza	20
65	Bey	Sami	20
365	Amouri	Brahim	21

Cet exemple représente une relation décrivant des étudiants. Etudiant est le nom de la relation. Les entêtes des colonnes NoEtud, Nom, Prénom et Age sont les **attributs** de la relation. Chaque ligne de la table correspond à une occurrence. Par exemple, <203, ghichi, faiza, 20> constitue un **tuple** qui décrit une occurrence.

Il est usuel de souligner l'attribut (ou les attributs qui constituent l'**identifiant** ou **clé** de la relation.

Cours

NomC	**Horaire**	**Prof**
Algo	Samedi08-10	Bentounsi
système	Mardi 10-12	bouzenada

suit

NoEtud	**NomC**

365	Algo
65	système
125	Algo
125	**système**

On remarque que l'identifiant de la relation « suit (qui traduit un type d'association) est composé des identifiants des deux relations précédentes.

b) Notion de domaine

Un domaine est un ensemble de valeurs que peut prendre ub attribut. Un domaine peut définir un ou plusieurs attributs.

Exemple :

Domaine(Noetud) : INF + entier[0..300]

Domaine(nom): chaîne de caractère de longueur maximale 20

Domaine(âge) : entier dans [16..60]

Relation

Une relation est définie par :

- Son nom
- La liste de ses attributs qui est une liste de couples (attribut, domaine)
- Son identifiant
- Sa définition qui est représentée par un texte expliquant l'utilité de la relation.

c) Schéma de relation :

Un schéma de relation est composé de son nom, la liste de ses attributs et son identifiant. Un schéma de relation a la forme suivante : **Nom-relation (A1 :D1, A2 :D2, …….., An :Dn)**

Exemple :

Etudiant (Noetud : Domaine(Noetud), nom : Domaine(nom), prénom : domaine(nom), age : domaine(age))

d) Population de relation :

La population d'une relation est constituée par l'ensemble de ses tuples. Dans cet ensemble, il n'y a ni double ni ordre.

Remarque : on appelle souvent le schéma d'une relation son intention et sa population son extension.

e) Schéma d'une base de donnée :

On appelle schéma d'une base de données l'ensemble des schémas de ses relations. De même, on appelle population d'une base de données, l'ensemble des populations de ses relations.

2) Règles de modélisation des attributs :

Dans le modèle relationnel, les attributs sont tous simples et monovalués : toute valeur prise par un attribut pour un tuple donné est atomique et unique. Les notions d'attributs complexe, multivalué et facultatif n'existe pas dans le modèle relationnel.

a) Attribut complexe :

Si on veut ajouter l'attribut «adresse » dans la relation Etudiant avec : Adresse = numéro + rue +ville + code-postal + pays », cet attribut est complexe et ne peut être géré par le modèle relationnel.

Solution 1 : on considère l'attribut «adresse » comme une chaîne de caractères. Da,ns ce cas, l'utilisateur ne peut pas poser des questions sur la ville, le pays ou un autre attribut composant « adresse ». il devra lire l'adresse et chercher le nom de la ville, par exemple, lui même par programme.

Solution 2 : on ajoute les attributs suivants à la relation Etudiant :

- numéro
- Rue
- Ville
- Code-postal
- Pays

Dans ce cas le SGBD connaît le détail de l'adresse mais ne connaît pas la notion globale d'adresse.

La solution est choisie en fonction de l'utilisation de l'attribut.

b) Attribut multi-valué :

Si on veut mémoriser les différents prénoms des étudiants, deux solutions sont possibles :

Solution 1 : avoir dans la relation «Etudiant », plusieurs attributs Prénom1, Prénom 2,…….Le problème qui se pose est celui du nombre d'attributs «prénom » à mettre dans la relation (problème de perte d'espace mémoire et de perte d'information). C'est une mauvaise solution dont il convient d'éviter l'utilisation.

Solution 2 : on ne garde que les attributs Noetud, nom, age dans la relation Etudiant et on crée une autre relation supplémentaire « Etudiant-prénoms »

Etudiant-prénoms :

Noetud	**prénom**
inf2000	Mohammed
inf2000	Ali
inf2002	Ahmed

c) Attribut facultatif:

Certains SGBDs relationnels gèrent une valeur spéciale appelée valeur nulle et notée "?". Cette valeur peut être prise par tout attribut (sauf si dans le schéma, on spécifie le contraire). Elle signifie que le tuple n'a pas de valeur pour cet attribut.

Exemple : < INF20000, Mohammed, Ahmed, ? > n'a pas la valeur de l'âge.

Une telle solution implique un traitement spécial dans les LMDs.

Exemple de schéma relationnel :

DOMAINES :
D-nom :chaîne de caractères de longueur maximale 20
D-Noetud : INF + entier dans [0..300]
D-age : entier dans [16..60]
RELATION Etudiant :
ATTRIBUTS :
Noetud : D-Noetud Sans NULL
Nom : D-nom Sans NULL
Age : D-age
IDENTIFIANT : (Noetud)

DEFINITION : toute personne actuellement inscrite à l'institut.

RELATION Etudiant-prénoms

ATTRIBUTS :

Noetud : D-Noetud Sans NULL

Prénom :D-nom Sans NULL

IDENTIFIANT : (Noetud + Prénom)

DEFINITION : prénom des étudiants

3) Dépendances fonctionnelles et clés de relations

Les dépendances fonctionnelles, notées DFs, est une notion importante utilisée dans la normalisation des relations. La normalisation st le processus qui permet de construire des relations complètes et non redondantes.

3-1) Définition

Soit R(X, Y, Z) un schéma de relation où $X \cup Y \cup Z$ sont des attributs de R.

On note : $\mathbf{X} \xrightarrow{R} \mathbf{Y}$ et on dit X *détermine* Y dans R ou Y *dépend fonctionnellement* de X dans R

Si et seulement si : étant donné une valeur de X, il lui correspond une et une seule valeur de Y (X et Y sont des attributs ou des ensembles d'attributs de R).

Exemple :

Soit la relation **Gestion-Stock** (code-prod, code-dep, libellé-prod, Pu, Adr-dep, Qte-stock).

Les dépendances suivantes sont vraies :

Code-prod $\longrightarrow$ libellé-prod

Code-dep $\longrightarrow$ Adr-dep

Code-prod + code-dep $\longrightarrow$ Qte-stock

Les dépendances suivantes sont fausses :

Code-prod $\longrightarrow$ Qte-stock

Code-prod $\longrightarrow$ Adr-dep

3-2) Propriétés des dépendances fonctionnelles :

Des axiomes et des règles permettent de déduire de nouvelles dépendances fonctionnelles à partir d'un ensemble de dépendances fonctionnelles initial. Dans ce qui suit, on désigne « A » l'ensemble des attributs de la relation R.

a) Axiomes d'Armstrong :

- *Réflexivité* : $Y \subseteq X \wedge X \subset A \Rightarrow X \xrightarrow{R} Y$
- *Augmentation* : $X \xrightarrow{R} Y \Rightarrow X\,Z \xrightarrow{R} Y\,Z$
- *Transitivité* : $X \xrightarrow{R} Y \wedge Y \xrightarrow{R} Z \Rightarrow X \xrightarrow{R} Z$

b) Règles d'inférence ou dérivées :

- *Union* : $X \xrightarrow{R} Y \wedge X \xrightarrow{R} Z \Rightarrow X \xrightarrow{R} Y \cup Z$
- Pseudo-*transitivité* : $X \xrightarrow{R} Y \wedge Y\,W \xrightarrow{R} Z \Rightarrow X\,W \xrightarrow{R} Z$
- *Décomposition* : $X \xrightarrow{R} Y \wedge Z \subseteq Y \Rightarrow X \xrightarrow{R} Z$

3-3) Types de dépendances fonctionnelles

a) DF triviale

Une DF $X \xrightarrow{R} Y$ est dite triviale si $Y \subseteq X$

$\longrightarrow$

b) DF canonique

Une DF $X \xrightarrow{R} Y$ est dite canonique si sa partie droite ne comporte qu'un seul attribut (simple) et un ensemble de DFs est dit canonique si chacune de ses DFs est canonique.

Exemple :

La DF code-prod libellé, Pu n'est pas canonique.

On applique la règle de décomposition

Code-prod $\longrightarrow$ libellé

Code-prod $\longrightarrow$ Pu

c) DF élémentaire

Une DF canonique $X \xrightarrow{R} Y$ est élémentaire si pour tout $X' \subset X$, la DF $X' \xrightarrow{R} Y$ n'est pas vraie. En d'autres termes, Y ne dépend pas fonctionnellement d'une partie de X

La DF Code-prod, code-dep $\longrightarrow$ libellé n'est pas élémentaire car Code-prod $\longrightarrow$ libellé

d) DF directe

Une DF $X \xrightarrow{R} Y$ est dite directe si :

- Elle est élémentaire
- Y ne dépend pas transitivement de X

3-4) Clés de relation :

a) Clé d'une relation : on dit qu'un attribut ou une liste d'attributs « X » est une clé pour la relation R(X,Y,Z) si $X \xrightarrow{R} Y \cup Z$

exemple : dans la relation : **Gestion-Stock** (code-prod, code-dep, libellé-prod, Pu, Adr-dep, Qte-stock), l'ensemble des attributs (code-prod + code-dep + libellé-prod) peut être une clé de Gestion-stock car elle détermine tous les autres attributs.

b) Clé minimale :

Une clé X est dite minimale si $X \rightarrow Y \cup Z$ est élémentaire.

Exemple : la clé (code-prod + code-dep + libellé-prod) n'est pas une clé minimale de Gestion-stock acar on a : code-prod + code-dep $\rightarrow$ Pu

La clé minimale est : (code-prod + code-dep)

c) Clé primaire/ clé candidate :

Si une relation possède plusieurs clés, on choisit une parmi elles qui sera appelée *clé primaire* ; les autres clés seront appelées *clés candidates* ou *clés secondaires*.

Exemple : soit la relation **Véhicule** (matricule, marque, type, Nochassis)

Dans cette relation, il existe deux clés matricule et Nochassis. On choisit par exemple « matricule » comme clé primaire et Nochassis comme clé candidate.

3-5) Graphe de dépendance fonctionnelle :

a) Graphe de DFs

Etant donnée une relation r et un ensemble de DFs, F, portant sur les attributs de R. F peut être représenté à l'aide d'un graphe dont les nœuds sont les attributs de R et les arcs sont les dépendances fonctionnelles elles même.

La construction d'un G.D.F se fait à partir d'un ensemble de DFs canoniques.

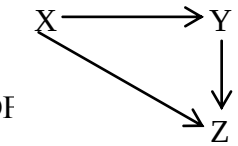
Exemple :

b) Graphe de DFs minimum :

Un G.D.F est dit minimum s'il ne contient pas de cycle, ie, il ne contient pas de DFs déduite par transitivité.

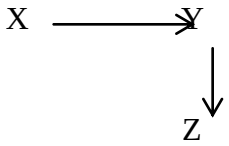
Exemple : Le G.D.F qui représente l'ensemble des DFs suivantes n'est pas minimum.

$F = \{X \rightarrow Y , \quad Y \rightarrow Z , X \rightarrow Z\}$



Le G.D.F qui représente l'ensemble des DF contre minimum.

$F = \{X \rightarrow Y , \quad Y \rightarrow Z \}$



Remarque : le G.D.F minimum peut être utilisé pour la recherche des identifiants des relations de la façon suivante :

Chercher sur le graphe, tout ensemble minimum d'attributs X tel que tous les chemins partant de X et suivant les arcs du graphe arrivent à tous les autres attributs du graphe. X est alors la clé ou l'identifiant de la relation.

4) La normalisation des relations

La normalisation utilise les deux notions de décomposition et de formes normales

4-1) La décomposition

a) Objectif : Etant donnée une relation non satisfaisante, c'est à dire redondante et posant des problèmes lors de la mise à jour des tuples, trouver un ensemble de relations satisfaisantes et qui décrive les mêmes informations.

La méthode consiste à décomposer la relation en deux relations (ou plusieurs). Pour cela, il est nécessaire de disposer de deux opérations : la projection et la jointure.

b) Définition de l'opération de projection :

Etant donnée une relation R(A1, A2, A3, ……,An), la **projection** de R sur un ensemble de ses attributs (Ai, ……), notée Π[Ai,…..]R, permet de créer une nouvelle relation d'attributs (Ai,…..) et de population égale à l'ensemble des tuples de R tronqués à (Ai,…..) **sans doubles.**

c) Définition de l'opération de jointure naturelle

Etant donné deux relations ayant au moins un attribut en commun, R1(X,Y) et R2(X, Z), la jointure naturelle de R1 et de R2, notée R1 ⊳⊲ R2 ou R1* R2, permet de créer 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 la même valeur pour X.

d) Définition de la décomposition

Une relation R (X, Y, Z) est <u>décomposable sans perte d'informations</u> (ie, sans perte d'attributs ni de DFs) en deux relations (ou plusieurs) R1= Π[X, Y]R et R2 = Π[X, Z]R si R = R1 ⊳⊲ R2

4-2) Les formes normales

Les trois premières formes normales et la forme normale de Boyce Codd ont pour objectif de permettre la décomposition des relations sans perte d'informations.

a) La première forme normale (1FN)

Une relation est dite en première forme normale, notée 1FN, si chacun de ses attributs a un domaine atomique et mono-valué.

Exemple : dans la relation **Personne** (Nss, nom, prénom, adresse), l'attribut adresse n'est pas atomique, il est composé des attributs No, rue, ville donc la relation n'est pas en 1FN.

b) La deuxième forme normale (2FN) :

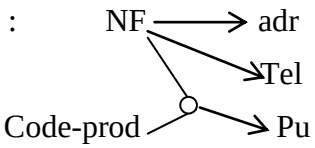
Une relation est dite en deuxième forme normale, notée 2FN, si et seulement si :

- Elle est en 1FN.
- Tout attribut n'appartenant pas à un identifiant (clé), ne dépend pas d'une partie de l'identifiant de la relation.

Exemple : soit une relation qui est en 1FN mais qui n'est pas en 2 FN :

Fournisseur1(NF, code-prod, adr, tel, pu)

Le graphe minimum des DFs de la relation :



Une telle relation pose des problèmes :

- *De redondance* : s'il existe 100 produits pour un fournisseur, on va répéter 100 fois son nom, son adresse et son téléphone.
- *De mise à jour pour les insertions* : quand on rajoute un produit, il faut rentrer à nouveau l'adresse et le téléphone du fournisseur.
- *De mise à jour pour les suppressions* : si on supprime la liste des produits d'un fournisseur, alors on supprime aussi le fournisseur.
- *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 2FN. La solution consiste à décomposer la relation en deux relations :

Fournisseur (NF, Adr, tel)

Catalogue (NF, code-prod, pu)

qui sont en 2FN.

c) La troisième forme normale (3FN) :

Définition 1 :

Une relation est dite en troisième forme normale, notée 3FN, si et seulement si :

- Elle est en 2FN.
- Tout attribut de la relation n'appartenant pas à un identifiant (clé), ne dépend pas d'un attribut non clé.

Définition 2 :

Une relation est en 3FN, si et seulement si :

- Elle est en 2FN.
- Tout attribut de la relation n'appartenant pas à la clé de la relation, ne dépend pas transitivement de celle-ci.

Exemple : soit une relation qui est en 2 FN : **Fournisseur2** (NF, pays, ville)

Il existe les DFs : NF $\longrightarrow$ ville et ville $\longrightarrow$ pays

(on suppose qu'il n' y a pas de villes homonymes dans la base)

NF $\longrightarrow$ pays est une DF déduite (par transitivité).

$\longrightarrow$ $\longrightarrow$

Graphe minimum de la relation : NF ville pays

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

Fournisseur (NF, ville)

Géo (ville, pays)

Remarque : on peut toujours décomposer une relation en 3 FN sans perte ni d'informations ni de DFs ; ce qui n'est pas le cas des formes normales plus poussées. D'où l'intérêt de cette forme normale.

d) La forme normale de Boyce Codd :

Bien qu'une relation soit en 3 FN, elle peut présenter des redondances crées, en général, par des dépendances d'attributs non clé vers une partie de la clé, d'où l'introduction de la forme normale de Boyce Codd (FNBC)

Définition :

Une relation est en FNBC si et seulement si :

- Elle est en 1FN,
- Les seules DFs élémentaires qu'elle comporte sont celles où clé détermine un attribut.

D'une façon générale, deux cas typiques sont à distinguer :

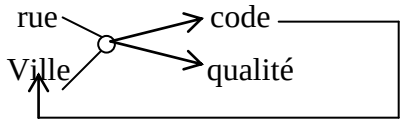
Premier cas : R (X, Y, Z) avec $X \longrightarrow Y$, $X \longrightarrow Z$, $\exists X' \subset X$ tel que $Y \longrightarrow X'$

Exemple :

Soit la relation : localisation (rue, ville, code, qualité)

Avec : rue, ville $\longrightarrow$ code
rue, ville $\longrightarrow$ qualité
code $\longrightarrow$ ville

Graphe de DFs :



Localisation

rue	ville	code	qualité
A.Ramdane	Constantine	251	Bon
L.Benmhidi	Constantine	251	Excellent
M.Didouche	Constantine	251	Moyen
A.Ramdane	Skikda	211	Bon
Y.Zighoud	Skikda	211	Moyen

Selon la définition de la FNBD, la décomposition serait comme suit :

R1 (rue, ville, qualité) avec rue + ville $\longrightarrow$ qualité

R2 (code, ville) avec code $\longrightarrow$ ville

On remarque que : la DF rue + ville $\longrightarrow$ code est perdue

Cependant il est possible de recomposer la relation par jointure.

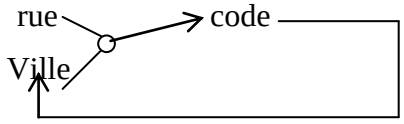
Deuxième cas : R (X, Y) avec $X \longrightarrow Y$, $\exists X' \subset X$ tel que $Y \longrightarrow X'$

Exemple :

Soit la relation : localisation (rue, ville, code)

Avec : rue, ville ⟶ code
code ⟶ ville

Le même problème de redondance se pose ; pour l'éliminer, on fait recours à la décomposition suivante :

Graphe de DFs : 

R1(code, rue)

R2 (code, ville) avec code ⟶ ville

On remarque que : la DF rue + ville ⟶ code est perdue

Exemple :

R1

Rue	code
A.Ramdane	251
A.Ramdane	252

R2

Code	ville
251	constantine
252	constantine

D'où A.Ramdane + constantine ⟶ 251, 252

Remarque

Il n'y a pas de solution idéale dans ce cas. Pratiquement, on conserve la relation initiale et on ajoute une contrainte d'intégrité spécifiant le fait qu'un code correspond à une et une seule ville.

5) Algorithme de décomposition en 3FN :

Pour décomposer une relation non normalisée en un ensemble de relations normalisées, un algorithme de décomposition en 3FN est proposé :

5-1) Quelques définitions de base :

a) Relation universelle :

C'est une relation composée de tous les attributs du domaine d'étude.

b) Fermeture transitive :

C'est un ensemble de DFS élémentaires enrichi de toutes les DFs élémentaires déduites par transitivité.

c) Couverture minimale :

C'est un ensemble «F» de DFs élémentaires associé à un ensemble d'attributs vérifiant les propriétés suivantes :

- Aucune DF dans F n'est redondante, ie, pour toute DF f de F, F - f n'est pas équivalent à F
- Toute DF élémentaire de F est dans la fermeture transitive de F.

Remarque 1 : deux ensembles F1 et F2 de DFs élémentaires sont équivalents s'ils ont la même fermeture transitive.

Remarque 2 : tout ensemble de DFs a une couverture minimale qui n'est en général pas unique.

5-2) Principe de l'algorithme :

Partant de la relation universelle et de ses DFs, le principe de l'algorithme est le suivant :

1. Trouver une couverture minimale de DFs, par exemple F.
2. Regrouper les attributs isolés (ceux qui ne sont ni source, ni cible dans aucune DF) au sein d'une seule relation dont tous les attributs sont clés.
3. Tant que l'ensemble des DFs est non vide faire
   - Rechercher le plus grand ensemble d'attributs X1, X2, ……, Xn dépendant fonctionnellement de X, X étant un attribut ou un ensemble d'attributs.
   - Constituer une relation d'attributs (X, X1, X2, ……, Xn) ayant X comme clé.
   - Eliminer les DFs X ⟶ X1,…….X⟶ Xn de F
   - Eliminer les attributs isolés.

Fin tant que

5-3) exemple :

soit la relation universelle : RU (code-prod, code-dep, libellé, pu, adr-dep, qte-stock)

et la couverture minimale :

F = {cod-prod $\overset{f1}{\rightarrow}$ libellé, code-prod $\overset{f2}{\rightarrow}$ pu, code-dep $\overset{f3}{\rightarrow}$ adr-dep, code-prod + code-dep $\overset{f4}{\rightarrow}$ qté-stock}

Application de l'algorithme :

1. Pas d'attributs isolés.
2. Création de la relation **stock(**code-prod, code-dep,qté-stock) à partir de la DF f4.
3. Elimination de la DF f4 et de l'attribut qte-stock
4. Création de la relation **produit** (code-prod, libellé, pu) à partir des DFs f1 et f2
5. Elimination des DFS f1 et f2 et des attributs isolés code-prod, pu et libellé
6. Création de la relation **dépôt**(code-dep, adr-dep) à partir de la DF f3
7. Elimination de la DF f3 et des attributs isolés code-dep et adr-dep
8. l'ensemble des DFs devient vide ➔ fin de l'algorithme.

la décomposition en 3 FN de la relation RU est :

produit (code-prod, libellé, pu)

dépôt(code-dep, adr-dep)

stock(code-prod, code-dep,qté-stock)

6) Les dépendances multivaluées et la quatrième forme normale

Certaines relations sont en 3 FN et même en forme normale de Boyce Codd, mais contiennent des redondances d'informations qui créent des problèmes lors de leur mise à jour.

Exemple :

Soit la relation **catalogue** (No-article, taille, couleur) qui décrit les habillements d'un grand magasin. On suppose que taille et couleur sont indépendants.

catalogue

No-article	taille	couleur
Pull(001)	38	Bleu
Pull(001)	40	Bleu
Pull(001)	42	Bleu
Pull(001)	44	Bleu

Pull(001)	38	Vert
Pull(001)	40	Vert
Pull(001)	42	Vert
Pull(001)	44	Vert

Cette relation n'⊂a pas de DF. L'identifiant est (No-article+taille+couleur)

Elle est en FNBC mais contient des redondances. Cet exemple montre l'insuffisance de la notion de DF qui ne permet pas de saisir l'indépendance qui existe entre des attributs tels que No-article et taille. Pour cela, on généralise le notion de dépendance fonctionnelle en introduisant celle de dépendance multivaluée (DM).

6-1) Dépendance multivaluée

a) Définition

Soit une relation R(X, Y, Z), Z peut être vide, il y a dépendance multi-valuée (DM) notée X⟶>> Y si à toute valeur de X correspond un ensemble de valeurs de Y qui est totalement indépendant de Z.

Remarque : les DF sont un cas particulier de DMs

Exemple : No-article ⟶>> taille

b) D.M élémentaire :

Une D.M X⟶>> Y est dite élémentaire si :

- Y n'est pas vide et est disjoint de X.
- La relation ne contient pas une autre D.M du type X' ⟶>> Y' telle que X'⊂X et Y'⊂Y.

Attention : la partie droite d'une D.M n'est pas nécessairement constituée d'un seul attribut (contrairement à une DF).

Exemple : Noemployé ⟶>> diplôme, année-diplôme

Avec : Noemployé ⇸>> diplôme

Noemployé ↛>> année-diplôme

C) Propriétés des D.Ms :

Les axiomes des D.M sont les suivants. Considérons une relation R composée d'un ensemble d'attributs A

Complémentation : X ⟶>>Y ⇒ X ⟶>> A- X -Y

Augmentation : X ⇸>> Y et V⊆W ⇒ XW ⇸>>YV

Transitivité : X ⟶>> Y ∧ Y ⟶>>Z ⇒ X ⟶>> Z -Y

D'autres règles sont déduites telle que la règle d'union.

Union : X ⟶>> Y ∧ Y ⟶>>Z ⇒ X ⟶>> Z -Y

6-2) Quatrième forme normale

La quatrième forme normale (4FN) est une généralisation de la FNBC afin de décomposer les relations ayant des D.M élémentaires.

a) Définition : on dit qu'une relation est en 4 FN si les seules D.M élémentaires sont celles dans lesquelles une clé détermine un attribut.

b) Exemple : dans l'exemple du catalogue (No-article, taille, couleur), on a les D.M suivantes :

No-article ⟶>> taille

No-article ⟶>> couleur

Cette relation n'est donc pas en 4 FN, on la décompose en deux relations :

Article-couleur (No-article, couleur)

Article-taille (no-article, taille)

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

Conclusion du chapitre

Une décomposition en plusieurs relations est bonne si

- Elle est utile :
  - Elle crée des relations décomposées plus normalisées que la relation initiale,
  - Certaines DFs ne doivent pas générer de décomposition : celles entre les attributs qui ne sont pas (ou très rarement)modifiés et qui sémantiquement ne méritent pas d'être décrits par une relation.
- 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.
- Il n'y a pas de perte de dépendance lors de la décomposition : toutes le dépendances non déduites doivent être dans l'ensembles des relations issues de la décomposition.