
Chapitre 4 : Normalisation

I. Introduction

La normalisation peut être effectuée, et c'est préférable, pendant la phase de conception sur le modèle entités-associations (voir chapitre2). Dans le cas où la normalisation est faite en amont, lors de la conception, il n'est pas nécessaire de la recommencer sur le modèle relationnel. On peut tout de même vérifier que les relations obtenues par le passage du modèle entités-associations au modèle relationnel sont toujours en forme normale, mais, sauf erreur, il ne devrait pas y avoir de problème. Il en va tout autrement lorsque l'on ne connaît pas bien, ou maîtrise pas bien, l'origine d'un modèle relationnel. Dans ce cas, vérifier la normalisation des relations, et, le cas échéant, les normaliser, est une phase primordiale.

Dans ce chapitre, on va voir la notion de la dépendance fonctionnelle ainsi que les différentes formes normales dans le but de normaliser une base de données à partir de la décomposition de la table mère en des sous tables.

II. Normalisation

1. Définition

Normaliser une relation consiste à la représenter sous une forme qui respecte certains critères afin de s'assurer de l'intégrité des données. La normalisation est un moyen fondamental mis à la disposition du concepteur de la base pour éviter un bon nombre d'erreur. Elle peut être définie comme le processus qui consiste, à partir d'une table universelle composée de la table des attributs, à appliquer sur laquelle un algorithme de normalisation pour avoir plusieurs tables.

Cet algorithme de normalisation est constitué des étapes qui vérifient si les tables dans des états bien définis. Ces états s'appellent : les formes normales (FNs).

2. Dépendances fonctionnelles

2.1. Définition

Lors de la définition d'un schéma conceptuel dans le modèle relationnel, le choix des relations est primordial. Ce choix doit être guidé par les dépendances qui existent entre les données, dans la réalité et qui sont des contraintes que doivent vérifier les données effectivement dans la base. Une dépendance est une contrainte exprimée par une règle qui doit vérifier les données afin de s'assurer que le schéma relationnel défini est correctement construit et que la BD soit dans un état cohérent.

2.2.Principe

Un attribut (ou groupe d'attributs) B d'une table T est fonctionnellement dépend d'un autre attribut A d'une table T, si à tout instant, chaque valeur de A n'a qu'une valeur associée de B.

On dit alors que A "détermine" B, et on note : $A \rightarrow B$

Exemple 1

Soit le schéma de relation PERSONNE (No_SS, Nom, Adresse, Age, Profession).

Les dépendances fonctionnelles qui s'appliquent sur ce schéma de relation sont les suivantes :

- No_SS \rightarrow Nom,
- No_SS \rightarrow Adresse,
- No_SS \rightarrow Age,
- No_SS \rightarrow Profession.

On pourra aussi écrire : No_SS \rightarrow Nom, Adresse, Age, Profession.

L'attribut No_SS détermine tous les attributs du schéma de relation. Il s'agit d'une propriété de la clé d'un schéma de relation.

Exemple 2

Considérons la relation suivante

PRODUIT (RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)

- Elle est visiblement redondante

RefProduit	LibelleProduit	PU	Quantité	NumService	Adresse	Capacité
P1	CH7	23.510	300	S1	Sousse	9000
P1	CH7	23.510	500	S2	Tunis	6000
P3	VIS12	0.150	900	S4	Sousse	2000

Cette relation présente certaines anomalies lors des manipulations :

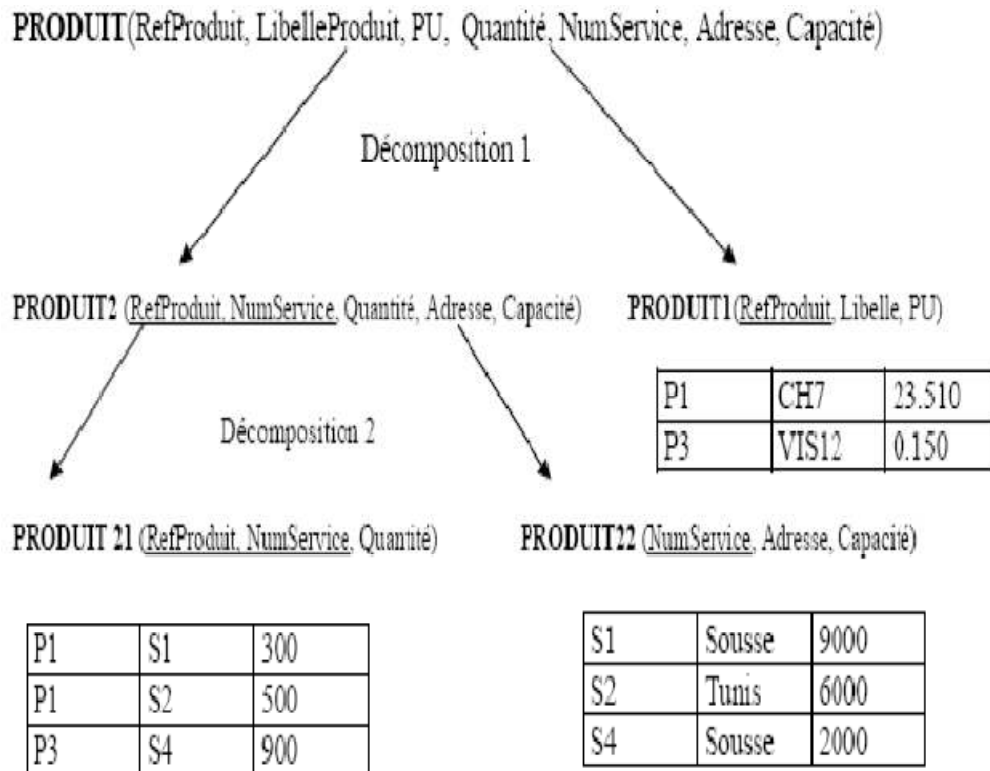
- **Redondance** : un produit apparaît autant de fois qu'il sera livré par un service.
- **Mise à jour** : faute de redondance, les mises à jour conduiront à des risques d'incohérence et de non intégrité.
- **Insertion et suppression** : l'insertion et la suppression ou le transfert d'attributs pourront faire apparaître des valeurs nulles.

On peut dire qu'une Base de Données relationnelle est 'correcte' ou normalisée si :

- ✓ Chaque relation décrit une information élémentaire avec les seuls attributs qui lui sont directement liés.
- ✓ Il n'y a pas de redondance d'information qui peuvent produire des problèmes de mise à jour.

La relation Produit peut être décomposée en trois relations non redondantes :

PRODUIT(RefProduit, LibelleProduit, PU, Quantité, NumService, Adresse, Capacité)



Le résultat final de la décomposition est donc les relations suivantes :

- **PRODUIT1** (RefProduit, LibelleProduit, PU) : contient les données relatives aux produits.
- **PRODUIT21** (RefProduit#, NumService#, Quantité) : contient les données relatives aux produits distribués par des services.
- **PRODUIT22** (NumService, Adresse, Capacité) : contient les données relatives aux services.

Ainsi, la robustesse de la nouvelle solution permettra :

- de minimiser la tâche de modification de la structure lorsqu'on désirera la mettre à jour avec de nouveaux éléments,
- Une meilleure compréhension des interrelations entre les données,
- d'éviter la redondance (intégrité ainsi que contre-performance liée),
- d'éviter les anomalies transactionnelles : insertion, suppression et de mise à jour,

- De faciliter l'écriture des requêtes non prévus au moment de la conception.

La normalisation est donc le processus de transformation d'une relation ayant des problèmes lors de la mise à jour vers une autre relation n'ayant pas ces problèmes.

Ci-dessous, un schéma définissant le processus de normalisation qui consiste qu'à partir des attributs regroupés dans une table mère, et à partir des dépendances fonctionnelles qui constituent la sémantique entre les attributs, on va appliquer un algorithme de normalisation qui est réellement les formes normales pour arriver à plusieurs tables ce qu'on appelle des tables normalisées (voir fig.1).

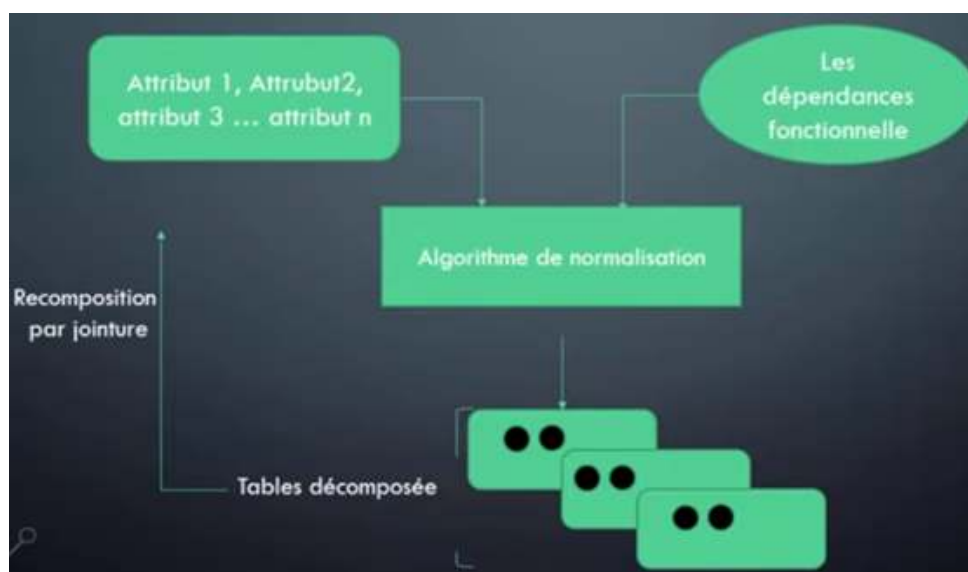


Fig.1: Processus de normalisation

2.3.Types de dépendance fonctionnelle

❖ Dépendance fonctionnelle élémentaire

Une Dépendance fonctionnelle $X \rightarrow Y$ est élémentaire si pour tout $X' \subset X$ la dépendance fonctionnelle $X' \rightarrow Y$ n'est pas vraie.

En d'autres termes, Y ne dépend pas fonctionnellement d'une partie de X (X est la plus petite quantité d'information donnant Y).

Exemple

RefProduit, LibelleProduit \rightarrow PU

n'est pas élémentaire car il suffit d'avoir la référence du produit pour déterminer le prix unitaire.

❖ Dépendance fonctionnelle canonique

- Une Dépendance fonctionnelle $X \rightarrow Y$ est canonique si sa partie droite ne comporte qu'un seul attribut.
- et un ensemble F de dépendances fonctionnelles est canonique si chacune de ses dépendances est canonique.

❖ Clé d'une relation

- C'est l'ensemble d'attributs dont les valeurs permettent de caractériser les n-uplets de la relation de manière unique.

Formellement

Un attribut ou une liste d'attributs X est une clé pour la relation $R(X,Y,Z)$ si

- Y et Z dépendent fonctionnellement de X dans
 $R : X \rightarrow Y, Z$.
- et $X \rightarrow Y, Z$ est élémentaire.

3. Formes normales

Les formes normales peuvent être considérées comme les différents stades de qualité, tout en s'appuyant sur les dépendances fonctionnelles entre les attributs, qui permettent d'éviter la redondance dans les bases de données relationnelles afin d'éviter ou de limiter : les pertes de données, les incohérences au sein des données, l'effondrement des performances des traitements.

Formellement, les différentes formes normales sont :

- la 1^{er} forme normale : «*1NF*»;
- la 2^e forme normale : «*2NF*»;
- la 3^e forme normale : «*3NF*»;
- la forme normale de Boyce Codd: «*BCNF*»;
- la 4^e forme normale : «*4NF*»;
- la 5^e forme normale : «*5NF*»;
- la forme normale domaine clef : «*DKNF*»;
- la 6^e forme normale : «*6NF*».

Les formes normales 1 et 2 sont essentielles à un MR formel. Ainsi, très souvent, le processus de normalisation respecte uniquement les trois premières formes normales. Dans ce cours, on s'intéressera aux cinq premières formes.

3.1. Formes normales : 1FN

Une relation respecte la 1^{ère} forme normale si et seulement si :

- Tous les attributs sont atomiques (valeur scalaire, indivisible).

Impacts :

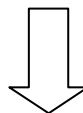
- Les attributs composites sont interdits;
- Les attributs multivalués sont interdits;
- Impose implicitement qu'un attribut membre de la clé primaire soit à valeur unique;
- Permet d'augmenter grandement la performance du SGBD en évitant de parcourir des attributs sous forme de liste et en créant des indexes performants.

Employé

<u>Id</u>	Nom	Supervise
3	Bernard, Alain	-
5	Labbé, Charles	3, 9
8	Barrette, Patricia	-
9	Cadieux, Violette	8



1FN



Employé

<u>Id</u>	Nom	Prénom	<u>Supervise</u>
3	Bernard	Alain	-
5	Labbé	Charles	3
5	Labbé	Charles	9
8	Barrette	Patricia	-
9	Cadieux	Violette	8

Attention redondance



1FN

3.2. Formes normales : 2FN

Une relation respecte la 2^{ème} forme normale si et seulement si :

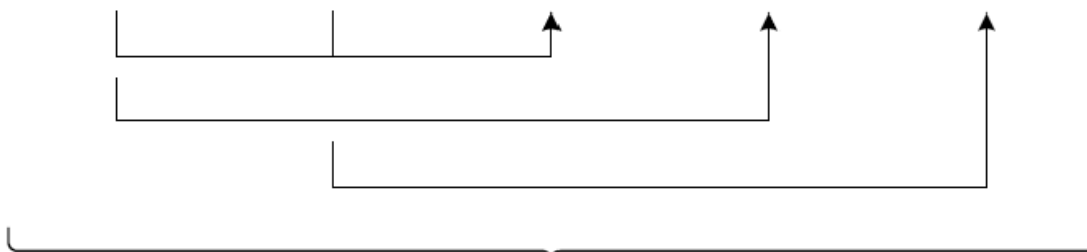
- Elle respecte la 1^{ère} forme normale;
- Tous les attributs *non clés* sont complètement dépendants de la clé primaire (ne peuvent dépendre que d'une partie de la clé primaire).

Impacts :

- Permet d'éliminer plusieurs redondances,
- Met en évidence certaines structures du monde réel.

ÉtudiantCours

<u>IdÉtudiant</u>	<u>IdCours</u>	CoteObtenue	NomÉtudiant	NomCours
3	12	B	Bernard	GPA775
5	15	A-	Labbé	GPA445
8	12	B+	Barrette	GPA775
5	17	A	Labbé	GPA789



2FN



ÉtudiantCours

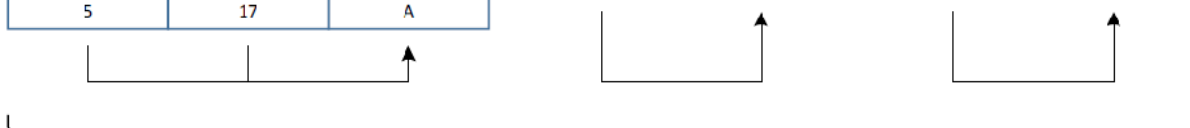
<u>IdÉtudiant</u>	<u>IdCours</u>	CoteObtenue
3	12	B
5	15	A-
8	12	B+
5	17	A

Étudiant

<u>IdÉtudiant</u>	NomÉtudiant
3	Bernard
5	Labbé
8	Barrette

Cours

<u>IdCours</u>	NomCours
12	GPA775
15	GPA445
17	GPA789



2FN

3.3. Formes normales : 3FN

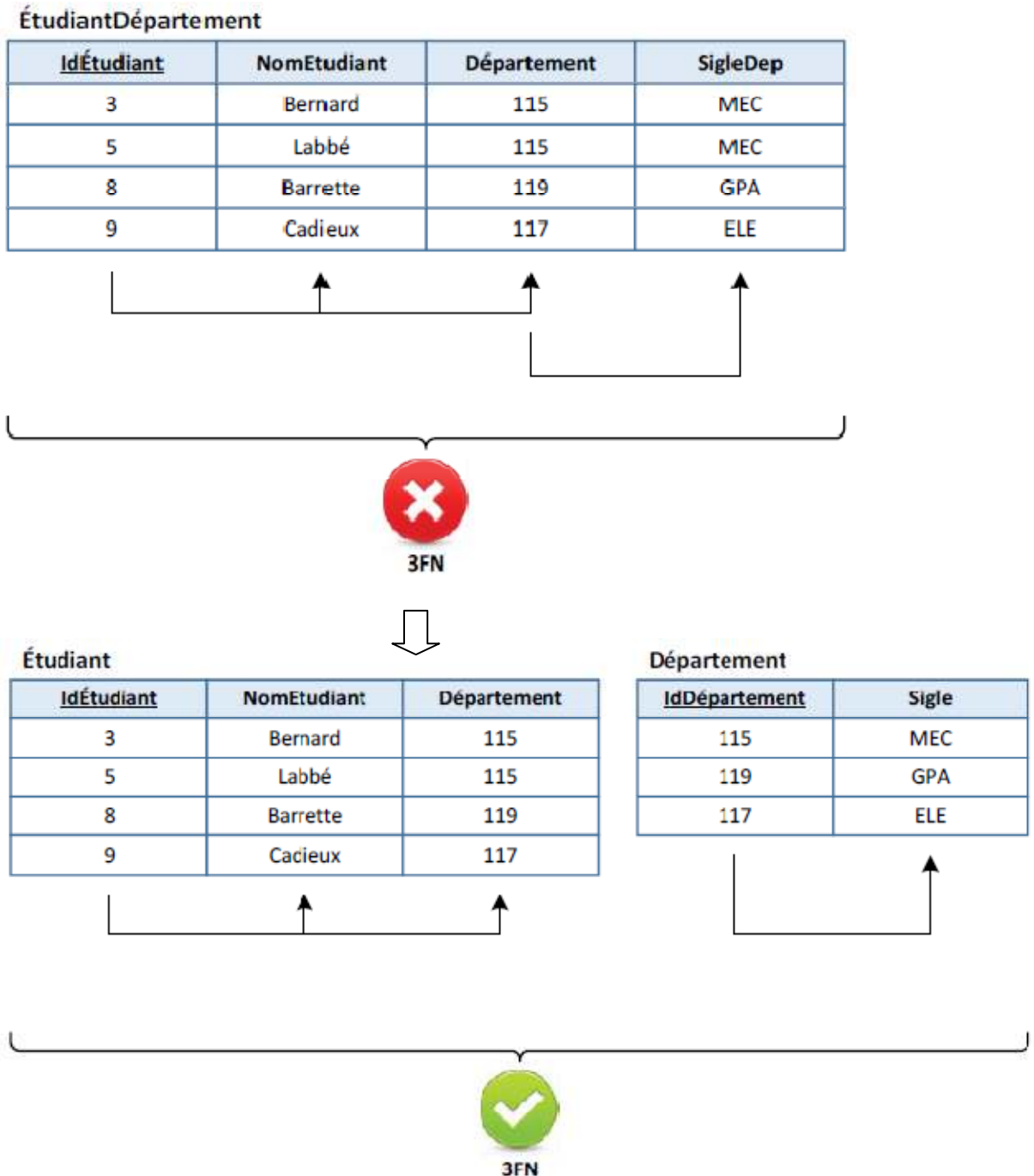
Une relation respecte la 3^{ème} forme normale si et seulement si :

- Elle respecte la 2^{ème} forme normale,

- Tous les attributs *non clés* sont dépendants non transitivement de la clé primaire (ne dépend pas d'un ou plusieurs attributs ne participant pas à la clé).

Impacts :

- Permet d'éliminer plusieurs redondances;
- Met en évidence certaines structures du monde réel;
- La 3FN est basée sur le concept de dépendance fonctionnelle transitive.



3.4. Formes normales : FNBC

Une relation respecte la forme normale de Boyce-Codd si et seulement si:

- Elle respecte la 3^{ème} forme normale;
- Tous les attributs *non-clé* ne sont pas source de dépendance vers une partie de la clé ().

